

Iñaki Prieto Furundarena

Gestión colaborativa de modelos 3D de ciudades en citygml durante su ciclo de vida basada en servicios en la nube

Departamento
Informática e Ingeniería de Sistemas

Director/es
BEJAR HERNANDEZ, RUBEN
IZKARA MARTINEZ, JOSÉ LUIS

<http://zaguan.unizar.es/collection/Tesis>



Reconocimiento – NoComercial – SinObraDerivada (by-nc-nd): No se permite un uso comercial de la obra original ni la generación de obras derivadas.

© Universidad de Zaragoza
Servicio de Publicaciones

ISSN 2254-7606



Universidad
Zaragoza

Tesis Doctora

GESTIÓN COLABORATIVA DE MODELOS 3D DE CIUDADES EN CITYGML DURANTE SU CICLO DE VIDA BASADA EN SERVICIOS EN LA NUBE

Autor

Iñaki Prieto Furundarena

Director/es

BEJAR HERNANDEZ, RUBEN
IZKARA MARTINEZ, JOSÉ LUIS

UNIVERSIDAD DE ZARAGOZA
Informática e Ingeniería de Sistemas

2018



Universidad
Zaragoza

tecnalia  Inspiring
Business

TESIS DOCTORAL

**GESTIÓN COLABORATIVA DE MODELOS 3D DE
CIUDADES EN CITYGML DURANTE SU CICLO
DE VIDA BASADA EN SERVICIOS EN LA NUBE**

Autor:

Iñaki Prieto Furundarena

Dirigida por:

Rubén Béjar, Jose Luis Izkara

UNIVERSIDAD DE ZARAGOZA

Informática e Ingeniería de Sistemas

2018

ÍNDICE

1	Introducción	1
1.1	Contexto general	1
1.2	Descripción general de la tesis.....	4
1.2.1	Ámbito de la investigación	4
1.2.2	Motivación.....	4
1.2.3	Principales contribuciones	5
1.3	Metodología de la investigación	6
1.4	Estructura del documento	8
2	Modelos 3D en la gestión y planificación colaborativa de las ciudades	11
2.1	Introducción	11
2.2	Servicios para la gestión y planificación de ciudades.....	14
2.3	Casos de estudio.....	16
2.3.1	Categorización del parque edificado en una ciudad	17
2.3.2	Plataforma colaborativa para la definición y seguimiento de un Plan Especial de Reforma Interior (PERI).....	26
2.3.3	Evaluación del impacto de aplicación de medidas de conservación de energía.....	34
2.4	Análisis de los casos de estudio	41
2.4.1	Comparación de los casos de estudio	41
2.4.2	Requisitos de los modelos 3D urbanos	43
2.4.3	Soporte para los requisitos identificados en las herramientas existentes	46
2.5	Conclusiones	49
3	Metodología de generación y actualización semiautomática de modelos CityGML.....	51
3.1	Introducción	51
3.2	Trabajo relacionado	52
3.2.1	Modelos de datos para ciudades en 3D.....	52

3.2.2	CityGML	58
3.2.3	Proyectos e iniciativas relevantes	61
3.2.4	Generación y mantenimiento de modelos CityGML.....	64
3.3	Ciclo de vida de ciudades en 3D.....	71
3.3.1	Workflow de generación.....	73
3.3.2	Operaciones de mantenimiento.....	84
3.4	Implementación	91
3.4.1	Generación.....	91
3.4.2	Operaciones de mantenimiento.....	95
3.5	Aplicación	97
3.5.1	Data model set up	98
3.5.2	Geometric generation.....	115
3.5.3	Semantic generation.....	124
3.5.4	Operaciones de mantenimiento.....	127
3.6	Discusión	128
4	Mantenimiento colaborativo y despliegue automático de modelos CityGML validados	131
4.1	Introducción	131
4.2	Trabajo relacionado	132
4.3	Problemática de la creación y mantenimiento colaborativo de modelos urbanos.....	134
4.4	Despliegue continuo de modelos CityGML	140
4.4.1	Despliegue continuo	140
4.4.2	Pipeline de despliegue de CityGML.....	141
4.4.3	Implementación de pipelines de despliegue de CityGML.....	146
4.5	Escenario de ejemplo	151
4.5.1	Configuración	152
4.5.2	Ejecución y resultados	156

4.6	Discusión	158
5	Conclusiones y trabajo futuro	161
5.1	Conclusiones	161
5.1.1	Metodología de generación y actualización semiautomática de modelos CityGML	161
5.1.2	Validación y despliegue colaborativo de modelos CityGML.....	162
5.2	Principales contribuciones	164
5.3	Trabajo futuro	166
6	Listado de méritos.....	169
6.1	Publicaciones del capítulo 2	169
6.2	Publicaciones del capítulo 3	169
6.3	Publicaciones del capítulo 4	170
7	Referencias.....	171

ÍNDICE DE FIGURAS

Figura 1-1 Modelo de investigación científica de Stokes [16].....	8
Figura 2-1 Interfaz gráfica de usuario de la herramienta de categorización	20
Figura 2-2 Visualización de la distribución de los parámetros en la herramienta de categorización.....	20
Figura 2-3 Distribución geográfica de las tipologías y los resultados de la categorización.....	24
Figura 2-4 Visualización de la vulnerabilidad de los edificios	25
Figura 2-5 Visualización de mapas de inundaciones a través de servicios WMS.....	25
Figura 2-6 Arquitectura de la plataforma U3DCloud	29
Figura 2-7 Workflow de plataforma colaborativa basada en servicios para la definición y seguimiento de un PERI.....	31
Figura 2-8 Plugin desarrollado en ArcGIS para la generación del modelo.....	32
Figura 2-9 Generación de un modelo BIM en Revit / Visualización del modelo IFC en el visor 3D	32
Figura 2-10 Recopilación de datos in-situ del edificio utilizando la tablet.....	33
Figura 2-11 Visualización de imágenes del edificio seleccionado.....	34
Figura 2-12 Diagrama UML para el ADE de Patrimonio Cultural	36
Figura 2-13 Diagrama UML para el ADE de Comportamiento Energético.....	37
Figura 2-14 Comparación de los resultados de los diferentes ECM	39
Figura 2-15 Visualización del impacto negativo de los ECM.....	40
Figura 2-16 Visualización del impacto positivo de los ECM	41
Figura 3-1 Los diferentes niveles de detalle en CityGML	59
Figura 3-2 Modelado coherente semántica y geométricamente en CityGML	60
Figura 3-3 Ejemplo de la ciudad de Lund modelado en CityEditor.....	66
Figura 3-4 Ejemplo de cómo generar un modelo CityGML en FME a partir de un fichero SHP.....	67
Figura 3-5 Resultado en CityGML utilizando la herramienta 3dfier (https://github.com/tudelft3d/3dfier).....	68
Figura 3-6 Ciclo de vida del modelo 3D urbano.....	72
Figura 3-7 Flujo de trabajo de generación del modelo 3D urbano.....	74
Figura 3-8 Create Buildings (LoD0, LOD1 and LOD2).....	81
Figura 3-9 Operación de concatenar varios modelos CityGML	85
Figura 3-10 Operación de reemplazar un edificio por otro en un modelo CityGML.....	86
Figura 3-11 Operación de seleccionar un conjunto de elementos de una zona del modelo CityGML	87
Figura 3-12 Completar el modelo CityGML con un modelo de un edificio más detallado	88

Figura 3-13 Operación de cambiar el tipo de elemento urbano	88
Figura 3-14 Ejemplo en el que se contemplan todas las posibles variantes	91
Figura 3-15 Proceso realizado para realizar el preprocesado entre parcelas y edificios	107
Figura 3-16 ADE a escala edificio	108
Figura 3-17 ADE building general.....	109
Figura 3-18 ADE building energy.....	109
Figura 3-19 ADE building KPI	110
Figura 3-20 ADE a escala distrito	111
Figura 3-21 ADE district environmental.....	111
Figura 3-22 ADE district social	112
Figura 3-23 ADE district KPI	113
Figura 3-24 ADE de otros elementos.....	114
Figura 3-25 Distrito de Santiago de Compostela	115
Figura 3-26 Zona de interés de Santiago de Compostela.....	116
Figura 3-27 Generar el modelo de un área de radio de 2 km alrededor del área de interés	117
Figura 3-28 Resultado de Santiago de Compostela en CityGML	120
Figura 3-29 Información en 2D de Frankfurt.....	122
Figura 3-30 Resultado de Frankfurt en CityGML.....	122
Figura 3-31 Zona de interés de Budapest.....	123
Figura 3-32 Resultado de Budapest en CityGML.....	124
Figura 3-33 Procesado geométrico de medianeras.....	125
Figura 3-34 Procesado geométrico de la orientación principal	126
Figura 4-1 Escenario hipotético	137
Figura 4-2 Pipeline de despliegue CityGML	142
Figura 4-3 Configuración del escenario de ejemplo	153
Figura 4-4 Visualización de UAT en Jenkins	155
Figura 4-5 Visualización de UAT en Cesium	155
Figura 4-6 Interacciones principales en el escenario de muestra	156
Figura 4-7 Diferentes desencadenantes del pipeline en Jenkins	158

ÍNDICE DE TABLAS

Tabla 2-1 Identificación de las tipologías. [30].....	23
Tabla 2-2 Requisitos de los modelos 3D urbanos	44
Tabla 3-1 Comparación entre diferentes modelos de datos de intercambio 3D [39]	53
Tabla 3-2 Características que pueden ser cambiadas en cada elemento urbano	90
Tabla 3-3 Datos requeridos para generar los modelos 3D urbanos de los casos de estudio	99
Tabla 3-4 Información utilizada para la generación de los elementos de transporte, mobiliario urbano, vegetación y zonas de agua del modelo de ciudad de Frankfurt.....	119
Tabla 4-1 Problemas identificados en el escenario hipotético	138
Tabla 4-2 Herramientas desarrolladas exclusivamente para la implementación del pipeline ...	148
Tabla 4-3 Librerías de terceros.....	150
Tabla 4-4 Tareas de cada fase en el marco del pipeline de validación.....	154

ACRÓNIMOS Y ABREVIATURAS

3D: 3 Dimensiones
ADE: Application Domain Extensions
AEC: Architectural, Engineering and Construction
ASC: ASCII Raster format
BIM: Building Information Modeling
CAD: Computer-Aided Design
CD: Continuous Deployment
CE: Casos de Estudio
CI: Continuous Integration
CityGML: City Geography Markup Language
COLLADA: COLLABorative Design Activity
CSV: Comma Separated Value
BD: Base de Datos
DSM: Digital Surface Model
DTM: Digital Terrain Model
DVCS: Distributed Version Control Systems
DXF: Digital Exchange Format
ECM: Energy Conservation Measures
GBXML: Green Building XML
GIS: Geographic Information System
GML: Geography Markup Language
HTML: HyperText Markup Language
IFC: Industry Foundation Classes
IGN: Instituto Geográfico Nacional
ISO: International Organization for Standardization
KML: Keyhole Markup Language
KPI: Key Performance Indicator
LIDAR: Laser Imaging Detection and Ranging
LOD: Level of Detail
OGC: Open Geospatial Consortium
OSM: Open Street Map
PERI: Planes Especiales de Reforma Interior
POI: Point of Interest
REST: Representational State Transfer
SHP: ESRI Shapefile

UAT: User Acceptance Test
UML: Unified Modeling Language
VRML: Virtual Reality Modeling Language
WFS: Web Feature Service
WMS: Web Map Service
VCS: Version Control System
XML: eXtensible Markup Language
XSD: XML Schema Definition

1 Introducción

1.1 Contexto general

La población mundial es predominantemente urbana. Más de la mitad de la población vive actualmente en zonas urbanas, el 54 por ciento en la actualidad y se prevé que para 2050 llegará al 66 por ciento, según datos de la ONU. Estas cifras ascienden al 75% en la actualidad cuando hablamos de Europa [1]. Esta situación genera nuevas problemáticas, necesidades y retos a resolver. Aparte del deterioro físico, uno de los mayores problemas a los que se enfrentan las ciudades es la pérdida de habitabilidad y calidad urbana. Por eso, entre los objetivos de las estrategias de gestión y conservación de las ciudades, debería incluirse la mejora de la calidad de vida de sus habitantes. En 2015, la ONU aprobó la Agenda 2030 sobre el Desarrollo Sostenible¹, una oportunidad para que los países y sus sociedades emprendan un nuevo camino con el que mejorar la vida de todos, sin dejar a nadie atrás. Entre los objetivos marcados hay uno directamente relacionado con la población en las ciudades (11 – Ciudades y Comunidades Sostenibles.).

Con respecto al consumo energético, en Europa el 40 % del mismo corresponde al sector de la construcción, siendo este el responsable de un 36 % de las emisiones de CO₂. [2] Esta situación está estrechamente relacionada con las emisiones provocadas por las edificaciones ya existentes. En Europa hay 50 millones de edificios con más de 50 años de antigüedad, y la mayor parte de ellos están ubicados en ciudades [3]. Además, los distritos urbanos representan un elemento de máxima importancia para la cultura y el patrimonio europeos, así que la conservación de sus edificios y la mejora de las condiciones de vida en ellos pueden contribuir decisivamente a proteger este legado para las futuras generaciones.

Las nuevas tecnologías aplicadas a las ciudades (*smart city*) están convirtiendo a las mismas en fuentes de grandes volúmenes de información muy dispar (diferente escala, uso, naturaleza, diferentes formatos, generados por diferentes agentes, etc.) que además se genera de manera distribuida, lo que la hace más difícil de mantener. Esta cantidad de información generada aumentará exponencialmente en el futuro, de modo que su

¹ <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

modelado, almacenamiento, procesado y actualización serán un aspecto crucial y estratégico de las labores de gestión y toma de decisiones en contextos urbanos [4].

El desarrollo de métodos innovadores y herramientas de gestión que sirvan para crear un enfoque sistémico y participativo a escala urbana puede ser esencial para explotar esta información. El análisis y procesamiento de la información con ayuda de las nuevas tecnologías puede ser de gran utilidad de cara a la priorización y a la toma de decisiones dentro de los procesos urbanos [5]. Por lo tanto, se hace necesario identificar y desarrollar herramientas que faciliten la gestión y el mantenimiento de la información para una gestión urbana efectiva en todos los niveles (eficiencia de los recursos, eficiencia energética, transparencia, respuesta rápida a contingencias, etc.).

Dentro de los retos del "Programa Estatal De I+D+i Orientada a los Retos de la Sociedad" [6], se identifican las 'Ciudades Inteligentes' dentro de los retos en economía y sociedad digital. La Comisión Europea también marca como objetivo del Horizon 2020 las 'Smart Cities and Communities - The European Innovation Partnership on Smart Cities and Communities' [7]. Las líneas temáticas claves dentro de las Smart Cities and Communities son las siguientes: sistemas urbanos inteligentes de energía y movilidad bajos en emisiones de CO₂; herramientas y servicios inteligentes para la integración de sistemas urbanos de energía y transporte; la gestión inteligente de datos "Smart Big Data", para transformar la información en conocimiento y mejorar la toma de decisiones relativa a la gestión de los sistemas urbanos y la gobernanza inteligente y participativa que acerque la tecnología al mercado y a los ciudadanos facilitando así entornos sostenibles y saludables.

Este escenario requiere el desarrollo de modelos de ciudades virtuales 3D para integrar información de diferentes escalas, fuentes y disciplinas en un único modelo coherente. Esta posición está respaldada por iniciativas y directrices europeas como E2B² (*Energy Efficient Buildings*) que se centra en modelos de información centralizada y abierta para mejorar el rendimiento energético de los edificios, la iniciativa reFINE³ (*Research for Future Infrastructure Networks in Europe*) del ECTP (European Construction Plataforma Tecnológica) donde se presentan a los modelos de datos como de vital importancia para la multimodalidad, la infraestructura y el transporte inteligente, y especialmente la

² <http://e2b.ectp.org/>

³ <http://ectp.ectp.org/TFI.asp>

directiva europea INSPIRE [8] cuyo objetivo es crear un marco jurídico que permita establecer una infraestructura de datos espaciales a nivel Europeo. La directiva INSPIRE permitirá disponer de más datos espaciales, más fiables y al alcance de todos. Otra directiva Europea relevante en el contexto de este trabajo es la directiva PSI- Public Sector Information (2003/98/CE- modificada 15 abril 2013), que establece el marco legal para la reutilización de la información del sector público por parte del sector privado con propósitos comerciales y no comerciales. Esta directiva reconoce además el valor de la publicación de datos abiertos, no sólo como valor económico, sino también como valor social y como tractor de la transparencia.

Varios proyectos europeos (por ejemplo, el proyecto EPIC⁴ - Plataforma Europea de Ciudades Inteligentes, y el proyecto SEMCITY⁵ - Enriquecimiento Semántico de modelos de ciudades 3D para el desarrollo urbano sostenible) llegaron a la conclusión de que es necesaria una mejor comprensión del sistema urbano para alcanzar los objetivos de desarrollo sostenible para las ciudades. Los modelos urbanos precisos en 3D son una herramienta importante para una mejor comprensión de los sistemas urbanos y, por lo tanto, para el desarrollo urbano sostenible.

Los modelos 3D urbanos son una representación digital, georreferenciada y tridimensional de los objetos, estructuras y fenómenos correspondientes a una ciudad desde una perspectiva multiescala [9]. Los modelos 3D urbanos incluyen información sobre el terreno, edificios, el uso del suelo, la vegetación y las carreteras, entre otros. Su principal característica es su capacidad de aunar datos sobre una ciudad en un único modelo de datos, simplificando así su uso e interoperabilidad [10]. Estos modelos son útiles para distintas aplicaciones, tales como la gestión de desastres, la planificación urbana y del tráfico, la seguridad, las telecomunicaciones, la navegación, la prevención de riesgos (p.ej. relacionados con la inundabilidad) o el turismo, entre otros [11] [12]. Su potencial se maximiza cuando incorporan información tanto geométrica como semántica [13].

⁴ <http://www.epic-cities.eu/>

⁵ http://www.cost.eu/COST_Actions/tud/TU0801

1.2 Descripción general de la tesis

1.2.1 Ámbito de la investigación

El objetivo de esta investigación es, por un lado, definir e implementar un flujo de trabajo para generar y mantener modelos 3D urbanos incluyendo, además de los edificios, otros elementos tales como zonas verdes, carreteras o puntos de interés. Por otro lado, se pretende facilitar el mantenimiento colaborativo de estos modelos 3D urbanos, abordando al mismo tiempo los problemas de validación y despliegue de los mismos.

El marco de referencia para esta tesis son los modelos 3D urbanos. Un modelo 3D urbano permite representar la información de los diferentes objetos urbanos georreferenciados. Además, resulta especialmente relevante disponer de información a diferentes escalas (edificio, entorno urbano, ciudad).

Al mismo tiempo, es necesario que dicho modelo combine información geométrica y semántica de forma coherente. El modelo debe ser único, y debe permitir presentar, manipular y gestionar los datos urbanos que luego pueden ser utilizados en diferentes aplicaciones.

Existen diferentes alternativas para la representación de modelos de datos a nivel urbano y a nivel de edificio. Sin embargo, para la unión conceptual de ambos niveles la opción más común es CityGML, que es el lenguaje de modelizado de datos elegido para los modelos 3D urbanos en esta investigación.

1.2.2 Motivación

El primer problema a resolver sobre los modelos 3D urbanos es que la generación de los mismos es compleja y costosa. Aunque la generación de la información geométrica de modelos 3D urbanos ha avanzado muchos en los últimos años, la semantización, que es el proceso de asignar información semántica a los objetos urbanos, sigue siendo uno de los aspectos críticos. Una vez existen, el siguiente problema a resolver es el mantenimiento, ya que normalmente los modelos se generan en un momento concreto, pero o no se actualizan o solo se actualizan cada mucho tiempo.

Otro de los problemas que han sido identificados después de años trabajando con modelos 3D urbanos y desarrollando aplicaciones basadas en los mismos, es la necesidad de validarlos. Cada vez que se realizan cambios en el modelo, más aún si estos cambios los

realizan diferentes personas, es necesario comprobar que el modelo sigue siendo válido. Normalmente la validación es un proceso semimanual que conlleva un gran coste en tiempo, y aun así no se tiene la certeza de que el modelo sea totalmente válido.

Ante esta situación es necesario, por un lado, diseñar y desarrollar metodologías para facilitar y agilizar la creación y mantenimiento de modelos 3D urbanos. Y, por otro lado, ofrecer soluciones que permitan trabajar conjuntamente de forma colaborativa en el mantenimiento de un modelo 3D urbano y ofrecer mecanismos para la validación automática de los mismos una vez actualizados. Con estas soluciones será más fácil disponer de modelos 3D urbanos actualizados, validados y puestos a disposición de sus usuarios fácilmente.

El objetivo de esta investigación es también avanzar en la correcta adopción de los modelos 3D urbanos con el fin de aumentar su uso en diferentes herramientas y por diferentes tipos de usuarios y organizaciones, tales como administraciones, empresas, universidades o ciudadanos.

1.2.3 Principales contribuciones

Se enumeran a continuación las principales contribuciones del trabajo realizado en esta investigación:

(1) **Diseño y desarrollo de una solución para el mantenimiento colaborativo y despliegue automático de modelos CityGML validados.**

Se presenta una solución para facilitar el mantenimiento regular de modelos 3D urbanos en CityGML. La solución se fundamenta en la estrategia y las herramientas de despliegue continuo empleadas en el desarrollo de *software* y se adapta al problema de crear, mantener y desplegar modelos CityGML validados.

(2) **Diseño y desarrollo de una metodología de generación y actualización semiautomática de modelos CityGML.**

Se implementa un flujo de trabajo que define las actividades a realizar para generar un modelo 3D urbano que incluya información geométrica y semántica de edificios y otros elementos urbanos en diferentes niveles de detalle. Se presentan, además, una serie de procesos que permiten realizar operaciones de mantenimiento sobre el modelo 3D urbano con el fin de facilitar su actualización.

(3) **Presentación de tres casos de estudio que motivan el uso de los modelos 3D urbanos.**

Se presentan tres casos de estudio que solucionan problemáticas diferentes teniendo como punto en común la utilización de un modelo 3D urbano. Estos casos de estudio sirven como motivación de utilización de modelos 3D urbanos.

(4) **Identificación de requisitos de los modelos 3D urbanos.**

Se han definido los requisitos de los modelos 3D urbanos analizando los tres casos de estudio presentados. Se han analizado también las herramientas existentes que permiten generar, editar y visualizar modelos 3D urbanos. De esta forma se definen algunos de los requisitos que debe cumplir un modelo 3D urbano para que sea reutilizable en diferentes ámbitos y aplicaciones.

1.3 Metodología de la investigación

Para lograr los objetivos de investigación expuestos en el apartado anterior, se ha realizado un trabajo en varias fases. La metodología de trabajo seguida en esta investigación resulta de una combinación de tareas analíticas, de diseño, desarrollo y evaluación. La metodología de referencia considerada durante el desarrollo de la tesis se basa en la denominada *Design Science Research* [14]. Esta metodología se basa en varios elementos:

- Relevancia del problema. La solución debe resolver un problema importante.
- Diseño como prototipo. Se deben producir prototipos viables y de interés.
- Rigor de la investigación. Se deben utilizar métodos rigurosos para la construcción y evaluación de los prototipos.
- Diseño como proceso de búsqueda. El diseño debe ser un proceso iterativo.
- Evaluación del diseño. Utilidad, calidad y eficacia deben ser demostradas.
- Contribuciones a la investigación. Las contribuciones pueden ser de diseño, fundamentos o metodológicas.
- Comunicación de la investigación. Se debe difundir la importancia del problema y la novedad de la solución.

La metodología de referencia se ha adaptado a las características del trabajo de investigación desarrollado durante la tesis. Los pasos seguidos en el desarrollo de la tesis han sido:

- Estudio del estado del arte: Por un lado, se han adquirido las habilidades y conocimientos técnicos necesarios para el desarrollo de la investigación. Además, ha sido necesario identificar las oportunidades y analizar y descomponer los problemas para proponer soluciones innovadoras. Esta labor ha sido más importante en las primeras fases del desarrollo de la tesis, pero se ha mantenido durante todo el tiempo que ha durado la misma.
- Integración en un equipo de expertos en el desarrollo de tecnologías y herramientas para el modelado de la información geoespacial: De esta forma es posible conocer de primera mano los problemas y la calidad de las soluciones actuales. Esta experiencia ha proporcionado oportunidades para identificar las áreas que requieren de mejores soluciones.
- Propuestas: En paralelo y de forma continuada se han identificado y diseñado propuestas de metodologías y soluciones tecnológicas para abordar los problemas identificados. Este proceso se ha llevado a cabo mediante un enfoque iterativo, con discusiones regulares con los otros miembros del grupo de investigación, permitiendo así refinar y mejorar las soluciones.
- Prototipos: Una vez que las propuestas de metodologías y soluciones tecnológicas se han perfeccionado lo suficiente, el desarrollo de prototipos ha permitido poner a prueba su viabilidad. Se ha mantenido contacto con expertos externos y potenciales usuarios de los resultados para poder entender mejor sus necesidades y se les han presentado los prototipos para que evalúen su utilidad.
- Publicaciones: Los resultados han sido publicados cuando se han obtenido resultados de interés, con el fin tanto de proporcionar experiencia en el tema y de recibir información adicional (revisión por pares, debates en talleres y conferencias, etc.)

En el proceso de la investigación centrado en la identificación de requisitos para los modelos 3D urbanos se ha seguido el método de investigación con estudio de casos, que permite validar los requisitos aplicándolos en casos de estudio reales [15].

Desde un punto de vista más general, la investigación llevada a cabo en esta tesis se puede considerar como “investigación básica basada en las necesidades sectoriales”. Strokes [16] defiende que la investigación científica no debe ser conceptualizada como un proceso lineal, sino más bien que las consideraciones de uso y el grado de comprensión fundamental buscado representan diferentes medidas frente a las cuales cualquier propuesta de investigación podría ser evaluada [17]. Basado en este argumento Strokes presenta su modelo de investigación científica basado en cuatro cuadrantes (ver Figura 1-1).

		Considerations of Use	
		Low	High
Quest for Fundamental Understanding	Yes	Pure Basic Research (Bohr)	Use-Inspired Basic Research (Pasteur)
	No	----	Pure Applied Research (Edison)

Figura 1-1 Modelo de investigación científica de Stokes [16]

Dentro de este modelo de investigación el cuadrante de arriba a la derecha hace referencia a la investigación básica inspirada en las necesidades de los usuarios, en ella convergen las necesidades de comprensión fundamental y las consideraciones de uso. La investigación que se puede clasificar dentro de este cuadrante no puede ser considerada como investigación puramente básica ni tampoco como investigación puramente aplicada, sino como una mezcla de ambas. Es en este cuadrante donde se enmarca el trabajo desarrollado en esta tesis.

1.4 Estructura del documento

Este documento presenta los resultados de la investigación llevada a cabo en esta tesis. El documento se organiza en 5 capítulos, cada uno de ellos presenta una estructura propia en función del tipo de información descrita. La estructura del documento presenta los pasos seguidos en la investigación.

El primero de los capítulos del documento (CAPÍTULO 1) describe la introducción a la tesis, el ámbito, la motivación, un resumen de las principales contribuciones y la metodología de investigación. Este capítulo representa la introducción y puesta en contexto de la investigación que se detalla en los siguientes capítulos.

El siguiente capítulo (CAPÍTULO 2) describe los modelos 3D aplicados a la gestión y planificación colaborativa de las ciudades. Este capítulo presenta una descripción de servicios aplicados para la gestión y planificación de ciudades. Después se presentan tres casos de estudio que solucionan problemáticas diferentes basándose en modelos 3D urbanos. A continuación, se identifican los requisitos de los modelos 3D urbanos y se describen las conclusiones.

A continuación (CAPÍTULO 3) se presenta una metodología que permite generar y actualizar modelos CityGML de manera semi-automática. Para ello, primero se describe el trabajo relacionado. Después se presenta el ciclo de vida de ciudades en 3D que se divide en el *workflow* de generación y las operaciones de mantenimiento. A continuación, se valida describiendo la implementación y aplicación de la metodología. El capítulo termina con la discusión.

En el CAPÍTULO 4 se presenta una solución para facilitar el mantenimiento regular de modelos 3D urbanos en CityGML. Para ello, primero se describe el trabajo relacionado. Después se presenta el concepto de despliegue continuo aplicado a modelos CityGML en el que se detalla la metodología a seguir. Después se valida la metodología en un escenario de ejemplo. El capítulo termina con la discusión.

El CAPÍTULO 5 presenta las principales conclusiones extraídas de la investigación descrita en este documento, así como las líneas futuras de continuación del trabajo realizado.

El documento concluye con una lista de referencias bibliográficas (libros, documentos, artículos y páginas web) que reflejan las fuentes de información consultadas durante la realización de la investigación y que proporcionan una importante fuente de recursos para conocer en mayor detalle algunos aspectos expuestos en esta tesis.

2 Modelos 3D en la gestión y planificación colaborativa de las ciudades

2.1 Introducción

La utilización de modelos 3D urbanos está adquiriendo gran relevancia en los últimos años ya que permiten el almacenamiento y visualización de información a escala urbana. El modelo de información urbano resulta de la aplicación del concepto de modelado de información del edificio (*Building Information Modeling* - BIM) al siguiente nivel de los espacios de vida humana, la ciudad. La visualización de modelos 3D y datos semánticos de ciudades o barrios se presenta como la manera más natural de presentar las propiedades espaciales de los elementos urbanos y es de gran utilidad para los diferentes agentes que participan en la gestión, conservación y uso de la ciudad.

El objetivo general de los modelos 3D urbanos es fusionar la información geométrica y las características relativas a los edificios y conjuntos urbanos en un único modelo de datos integrado. Se trata de desarrollar tecnología para crear modelos de información de la ciudad que puedan ayudar en la planificación, diseño y análisis de los diferentes aspectos de las ciudades. El modelo 3D urbano se centra exclusivamente en una pieza crítica de la tecnología que va a hacer que las ciudades inteligentes existan, pero que tiene que trabajar en conjunto con otras tecnologías como lo son el *big data*, internet de las cosas (IoT) o monitorización en tiempo real de sensores, entre otros.

Los modelos 3D urbanos son una solución cada vez más popular para la representación de la información urbana. Sin embargo, los modelos 3D urbanos tienen que: 1) ser realistas, 2) combinar información geométrica y semántica, 3) ser útiles para diferentes usuarios y 4) ser usables a través de diferentes herramientas. Además, dichos modelos deben enfrentarse a varios desafíos:

- Aunque existen muchas fuentes de datos de gran interés para la creación de estos modelos, suelen existir problemas de incompatibilidad de formatos y de modelos de datos; la documentación es por norma general muy escasa; puede haber cambios inesperados y las licencias de utilización de los mismos no suelen ser claras. Es por eso importante que los modelos 3D urbanos cumplan los estándares y formatos más habituales, tales como CityGML [10].

- El mantenimiento de estos modelos a lo largo del tiempo (actualización, corrección, ampliación) entraña una gran complejidad, acompañada de problemas técnicos, legales y financieros [18].
- Durante los últimos años, la tecnología web para manipular modelos tridimensionales ha dado un gran salto. Pero siguen existiendo obstáculos que superar y, además, las herramientas de manipulación de modelos 3D que integran información semántica no disponen de todas las funcionalidades y posibilidades necesarias [19].

La creación, el uso y el mantenimiento de modelos 3D urbanos con información geométrica y semántica sigue siendo un reto actualmente. Desde la creación de modelos de ciudad CityGML a partir de diferentes fuentes de datos, haciendo hincapié en la generación de edificios y otros elementos urbanos (tales como vegetación, carreteras o mobiliario urbano, entre otros); hasta la extracción automática de información semántica y la tan a menudo olvidada tarea de mantenimiento.

También hay que hacer hincapié en las diferencias de los modelos de diferentes ciudades, ya que pueden variar en 1) tamaño de la ciudad, lo que va directamente ligado en el número de edificios, 2) las fuentes de datos, que pueden ser tales como el catastro, *Open Street Map* (OSM) o *Computer-Aided Design* (CAD), entre otros, 3) detalle de la geometría, desde edificios geoméricamente simplificados hasta edificios en las que se detallan todas las geometrías exteriores y patios interiores.

De la misma forma, no es tan inusual que, incluso dentro de un mismo ayuntamiento, varios departamentos no compartan datos entre sí, creando modelos de ciudad nuevos y diferentes a medida que los van necesitando para su trabajo. Por este tipo de situaciones se pretende promover la creación de modelos 3D urbanos unificados y actualizados que sean útiles para diferentes actividades, contribuyendo así a reducir los costes derivados de las tareas de creación y mantenimiento de modelos.

Una manera de probar su potencial real (compatibilidad con estándares internacionales, interoperabilidad con otros modelos de datos y herramientas de análisis, gestión y toma de decisiones, etc.) es con el desarrollo de un sistema basado en servicios que facilite la planificación y gestión urbana por medio de la creación de aplicaciones en la nube [20]. Este sistema debe contemplar y emplear varios estándares que contribuyan a definir y sacar el máximo beneficio de un modelo común de datos. Entre estos servicios se incluye,

por ejemplo, la rehabilitación sostenible, para orientar la administración y el mantenimiento de las intervenciones urbanas hacia la mejora de la sostenibilidad y del ahorro energético en las ciudades.

Asimismo, también se pueden diseñar otros servicios dirigidos a garantizar la eficiencia en la gestión de los recursos energéticos para la administración y optimización de la movilidad urbana, información turística, cultural y de servicios que fomenten el gobierno electrónico y la participación del público general, mejorando así la interacción y la comunicación entre los órganos de gobierno y la población general, entre otros [21]. También existen aplicaciones similares para el análisis energético en ciudades [22], o la estimación de la demanda energética de los edificios [23], así como la clasificación automática de los tipos de edificios [24] o la calculadora de la huella de carbono en zonas urbanas [25].

Los modelos 3D georreferenciados se están reforzando como la primera elección para el almacenamiento y la visualización de datos urbanos en numerosas áreas de aplicación. CityGML es un modelo de datos y formato de intercambio estandarizado que presenta datos semánticos comunes para entidades de ciudades 3D y sus interrelaciones, que se ha convertido en una de las opciones más utilizadas para la gestión de este tipo de información. Hoy en día, crear y mantener estos modelos CityGML es difícil y costoso. Esto se debe, en parte, a que la generación de las geometrías y el volcado semántico son procesos de elevada complejidad que requieren de cierta intervención manual. De hecho, muchos de los modelos CityGML disponibles para el público contienen errores.

Más que la generación de la parte geométrica es en la semantización del modelo de ciudad 3D donde el desafío es mayor. Después de generar y volcar la información semántica necesaria al modelo de datos, se requiere un mantenimiento continuo para actualizarlo. Es preferible que el sistema se actualice, en lugar de crear uno nuevo para cada decisión a tomar. En el marco del mantenimiento, hay que actualizar tanto los datos semánticos, como los geométricos. Sin este proceso, el modelo será menos útil a corto plazo. Se llevó a cabo una investigación sobre los factores de obsolescencia y las estrategias para la prevención de la misma, estando las estrategias implementadas descritas en [18]. Las tareas de mantenimiento a menudo consisten en recabar datos históricos acumulados de diferentes fuentes y actualizar la base de datos con ellos [26]. Otra posibilidad es mantener el modelo de datos de forma automática, por medio de la reconstrucción por componentes

[27]. En este trabajo se combinó información proveniente de diferentes sensores y mapas de un sistema de información geográfica con un enfoque semántico y basado en componentes de cara al modelado y a la reconstrucción de edificios complejos. La geometría y semántica resultantes se aplican después a un sistema de información espacial que permite realizar consultas para el mantenimiento y actualización del modelo de datos.

2.2 Servicios para la gestión y planificación de ciudades

Una *smart city*, o ciudad inteligente, es una ciudad que aplica las tecnologías de la información y de la comunicación (TIC) con el objetivo de proveerla de una infraestructura que garantice un desarrollo sostenible, un incremento de la calidad de vida de los ciudadanos, una mayor eficacia de los recursos disponibles y una participación ciudadana activa.

Las *smart city* pueden requerir el desarrollo de herramientas que faciliten la gestión y planificación urbana a través de la generación de soluciones en la nube (*cloud computing*). Para ello es necesario un modelo único de datos que facilite poner en marcha estos servicios.

Esto puede conseguirse a través de la creación de un sistema de servicios para la *smart city* basado en un modelo único de información multiescala. Estas soluciones de *smart city* deben considerar y usar diversos estándares para ayudar a definir y aprovechar un modelo de datos común. Estas aplicaciones o servicios pueden abarcar todas las necesidades de la ciudad.

Por un lado, existen servicios genéricos aplicados a diferentes dominios, tales como la movilidad, sostenibilidad, turismo o *e-government*, entre otros [28]:

- Servicios para la **rehabilitación sostenible**. Se trata de servicios que faciliten a la administración la gestión y el mantenimiento de las intervenciones en el entorno urbano. Estas intervenciones estarán encaminadas a la mejora de la sostenibilidad y ahorro energético del entorno urbano en su conjunto.
- Servicios para la **gestión eficiente de recursos energéticos**. A través de una visión holística de la ciudad es posible priorizar los puntos y factores clave para la optimización de consumos, uso de renovables y gestión de la demanda.

- Servicios para la **gestión y optimización de la movilidad urbana**. Como soporte a la toma de decisiones y planificación de actuaciones en la ciudad. Dentro de estos servicios se consideran aspectos de transporte, seguridad y accesibilidad.
- Servicios de **información turística y cultural**. Provisión de servicios y aplicaciones que proporcionan información de la ciudad relacionada con información turística y/o eventos culturales. Estos servicios explotarán, entre otras, la componente 3D del modelo de información.
- Servicios que favorezcan el **e-government y la participación ciudadana**., facilitando la interacción y comunicación entre administración y ciudadanos. Estos servicios permitirán definir políticas y tomar decisiones sobre la base de la inclusión, sostenibilidad y participación.
- Servicios para la **gestión eficiente de servicios públicos e infraestructuras urbanas**. Aprovechando la información del modelo será posible disponer de un “cerebro” que permita generar indicadores e informes para mejorar los servicios.
- Servicios de **alerta temprana e identificación y gestión de problemas futuros**. Estos servicios permitirán la monitorización de diferentes sensores que permitan tener controlado cualquier problema.
- Servicios para **análisis temáticos urbanos** (energéticos, corrientes de aire, ruidos, visibilidad...). Estos servicios permitirán realizar cálculos/simulaciones temáticas que después puedan visualizarse para su interpretación.

Por otro lado, existen servicios que han sido desarrollados basándose en un modelo 3D urbano y que dan respuesta a una problemática concreta [11]:

- Servicios para la **estimación de irradiación solar y sombreado**. Estos servicios permiten estimar cómo de expuesto está un edificio al sol de cara a la instalación de paneles fotovoltaicos.
- Servicios para la **geo-visualización y visualización mejorada**. Estos servicios permiten aprovechar la información geométrica y semántica para facilitar la interpretación de los datos mediante visores 3D geolocalizados.
- Servicios que permiten representar el **catastro en 3D**. Estos servicios permiten registrar toda la información del registro de propiedades de un edificio, teniendo así en cuenta edificios geoméricamente complejos difícilmente representables en 2D.

- Servicios para **dar respuesta a situaciones de emergencia**. Estos servicios permiten la gestión de desastres y la respuesta de emergencia proporcionando información valiosa, como la ubicación de los puntos de entrada del edificio, ayudar a los bomberos a identificar las mejores rutas para llegar o calcular el número de personas que es necesario evacuar.
- Servicios que permiten realizar la **planificación urbana**. Estos servicios permiten facilitar tareas tales como el diseño de un parque o investigar que objetos urbanos interfieren con la planificación de una nueva línea de metro, entre otros.
- Servicios de **visualización 4D**. Permiten presentar la información basada en la información 3D del modelo incluyendo la componente temporal (pasado o futuro). Se trata de ofrecer una herramienta de visualización para facilitar la gestión y presentación de la información asociada al modelo de información 3D.

De los servicios descritos anteriormente, en los casos de estudio que van a detallarse a continuación se desarrollan los servicios: de análisis temáticos urbanos; que favorezcan el *e-government* y la participación ciudadana; para la gestión eficiente de recursos energéticos y para la rehabilitación sostenible.

2.3 Casos de estudio

A continuación, se describen tres casos de estudio que presentan problemáticas diferentes que se han abordado mediante la utilización de un modelo 3D urbano y el desarrollo de herramientas y servicios que se basan en dicho modelo. Los tres casos de estudio son desarrollos realizados en el marco de proyectos de investigación llevadas a cabo en Tecnalia y en los que se ha participado activamente.

2.3.1 Categorización del parque edificado en una ciudad

Contexto

La mejora de la eficiencia energética de los edificios es uno de los aspectos clave para la reducción de emisiones de contaminantes y la mejora de la sostenibilidad a escala mundial. La rehabilitación energética de edificios existentes está siendo una de las apuestas clave para la revitalización del sector de la construcción en Europa. Abordar dicha rehabilitación a escala urbana supone importantes beneficios frente a la rehabilitación individual de edificios, por ejemplo, permite identificar sinergias entre elementos urbanos para compensar restricciones o desequilibrios entre demanda y potencial de generación, también permite aprovechar los beneficios de la economía de escala. Sin embargo, también lo convierte en una tarea más compleja, especialmente porque la necesidad de información para abordar el problema a dicha escala se multiplica respecto a la escala de edificio y lo hace extremadamente complejo.

Por otro lado, en los últimos años la frecuencia de eventos de precipitación extrema y el aumento del nivel del mar han aumentado y han impactado en un gran número de áreas urbanas. Por eso, la preocupación por la mitigación de desastres relacionada con el cambio climático es cada día mayor. Las decisiones de planificación urbana implican una comprensión de las interacciones complejas entre los diferentes aspectos de la ciudad, en su sistema constructivo, social, económico, ambiental y cultural. El análisis de estas interacciones requiere un enfoque sistémico ya que los componentes operan en diferentes escalas espaciales y temporales y generan una gran cantidad de datos. La comprensión de la información puede ser apoyada y homogeneizada por un modelo urbano de multi-escala, para facilitar la comprensión de las interacciones y el vínculo entre las diferentes disciplinas involucradas. Esta información puede usarse para determinar la vulnerabilidad de las áreas urbanas mediante su evaluación a nivel de edificio, a través de la creación de tipologías que representan el inventario de edificios, a menudo caracterizado por similitudes y elementos constructivos comunes.

Para facilitar estas dos tareas es posible identificar las principales tipologías de edificios de un entorno urbano y la selección de un edificio representativo de cada tipología. De esta forma se simplifica la recogida de datos y se utilizan los datos de los edificios representativos como base para los cálculos teniendo en cuenta la representatividad de dicha tipología en el distrito. Esta solución contribuye a reducir drásticamente la

necesidad de recogida de datos. Al mismo tiempo, también ayuda a medir el impacto de las diferentes intervenciones en cada tipología específica teniendo en cuenta la representatividad de la tipología en cuestión.

Modelo

En este caso de estudio se ha elegido, por un lado, el distrito histórico de Santiago de Compostela para probar la metodología y herramienta de categorización del parque edificado. El área seleccionada incluye un total de 805 edificios, de los cuales 699 son residenciales y 106 no residenciales. Se ha generado un modelo CityGML del distrito histórico de Santiago de Compostela que incluye: (1) un modelo digital texturizado del terreno de la zona urbana de Santiago de Compostela; (2) la geometría de edificios en el distrito histórico en LOD2 (*Level Of Detail*); (3) el modelo de datos de CityGML ampliado con parámetros específicos de la rehabilitación energética de edificios; (4) otros elementos urbanos como zonas verdes y carreteras.

Por otro lado, se ha generado también el modelo 3D urbano de los edificios cercanos a la ribera del río Urumea de Donostia – San Sebastián. El modelo de Donostia se ha utilizado para dar respuesta a la mitigación de desastres naturales relacionadas con el cambio climático tales como la subida del nivel del mar. Este modelo CityGML incluye alrededor de 2200 edificios.

Las fuentes de datos empleadas para la elaboración de los modelos de ciudad 3D han sido:

- Planos de los edificios del catastro
- Planos de otros elementos urbanos del Instituto Geográfico Nacional (IGN) y OSM
- Modelo digital de terreno del IGN
- Ortofotos del IGN
- *Laser Imaging Detection and Ranging* (LIDAR) del IGN y GeoEuskadi

Mientras que algunos de los atributos necesarios ya forman parte de los datos fundamentales de CityGML (tales como uso principal, año de construcción o número de plantas), otros han tenido que ser añadidos al modelo de datos con posterioridad (tales como área, nivel de protección o número de fachadas) mediante una extensión. La información semántica ha sido completada con ayuda de información del catastro español

y el plan especial del distrito histórico en el caso de Santiago de Compostela; y con datos abiertos disponibles en GipuzkoaIreka para el caso de Donostia – San Sebastián.

En este caso de estudio los modelos CityGML resultantes han sido desplegados en base de datos para después poder utilizar servicios estándares de acceso tales como el *Web Feature Service* (WFS).

Aplicación

Se ha implementado una herramienta software que automatiza parte de las tareas de la categorización del parque edificado y ofrece al usuario una solución sencilla e intuitiva para identificar las tipologías de edificios. Esta herramienta ha sido desarrollada dentro del proyecto europeo EFFESUS⁶. Para ello, la herramienta se basa en los datos incluidos en un modelo 3D urbano en CityGML. Además, permite al usuario editar los parámetros semánticos, intervalos y umbrales para la categorización, así como seleccionar los edificios representativos y completar sus propiedades.

La interfaz gráfica de usuario de la herramienta de categorización se divide en 4 áreas o paneles principales (véase Figura 2-1). Cada panel consta de varias pestañas que se habilitan o deshabilitan en función de la fase del proceso de categorización.

La herramienta de categorización adaptada al caso de la mejora de la eficiencia energética en el centro histórico de Santiago de Compostela está disponible bajo la URL:

<http://3dcity.tecnalia.com/DynaCategorization/faces/DynaCategorization.xhtml?city=effesussantiago>

⁶ <http://www.fffesus.eu/>

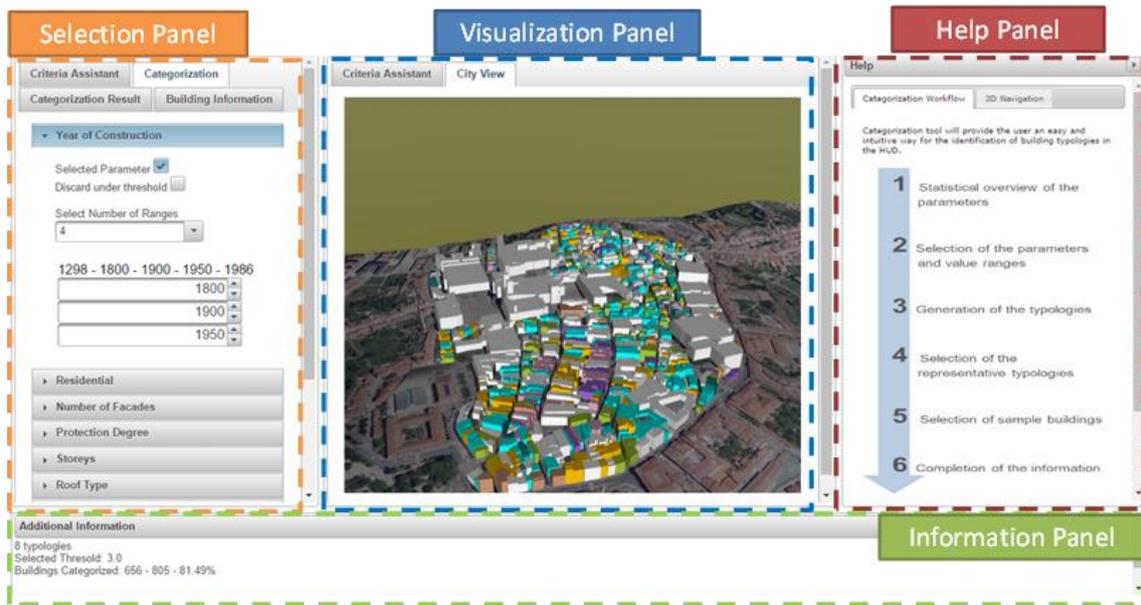


Figura 2-1 Interfaz gráfica de usuario de la herramienta de categorización

La herramienta de categorización dispone de un asistente que ofrece información sobre la distribución de los valores de los parámetros contenidos en el modelo, cuya finalidad es ofrecer información sobre la distribución estadística de los datos de los edificios en todo el distrito. Para ello se generan histogramas de frecuencia y *boxplot* automáticamente para cada parámetro (véase Figura 2-2).

Grado de protección

Año de construcción

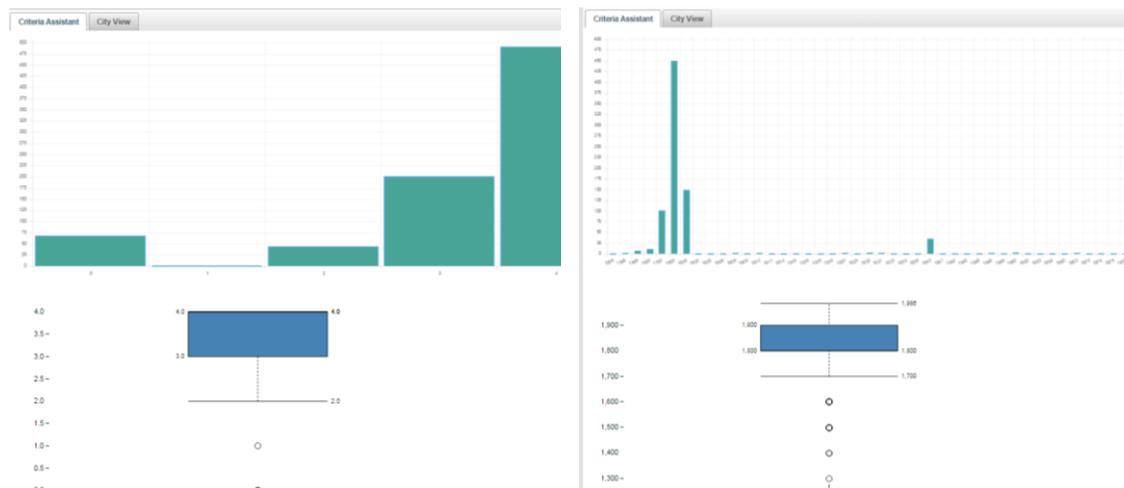


Figura 2-2 Visualización de la distribución de los parámetros en la herramienta de categorización

De cara a configurar los parámetros, intervalos y umbrales para la categorización, la herramienta presenta una funcionalidad que permite identificar automáticamente el número adecuado de intervalos y sus límites máximo y mínimo para cada parámetro. La configuración automática del número de intervalos y de los límites se basa en la adaptación del algoritmo *k-means* extendido para fines de agrupación [29]. El usuario debe seleccionar los parámetros y el umbral manualmente. También puede editar a mano los intervalos identificados automáticamente por la herramienta para adaptar los resultados a sus preferencias y conocimientos. Para ello, la herramienta obtiene la información de todos los parámetros desde el modelo CityGML y realiza el cálculo para obtener los intervalos adecuados.

Hay que tener en cuenta que solo se pueden definir intervalos para aquellos parámetros dotados de valores numéricos. En el caso de parámetros como el uso del edificio (p. ej. residencial, oficinas o comercial) no se define ningún intervalo y la herramienta de categorización contempla todos los valores aplicables.

Una vez identificadas todas las tipologías posibles de acuerdo con el número de parámetros e intervalos, se aplica el umbral para identificar las tipologías más representativas.

La configuración automática de los parámetros e intervalos para el proceso de categorización analiza cada parámetro de acuerdo con el siguiente procedimiento. En primer lugar, se estudia la distribución del parámetro. Si presenta menos de 5 valores diferentes, en la categorización se usan todos ellos. De lo contrario, se crean distintos intervalos en función de la distribución de los datos. Si ésta es homogénea (cada valor engloba menos del 20 % de los elementos), se crean 4 intervalos, cada uno con el 25 % de los valores. Si el 80 % de los elementos está englobado en un único valor, solo se crea un intervalo, que es el valor que más elementos tiene. Si no se configuran 1 ni 4 intervalos, entonces se aplica el algoritmo *k-means* para decidir si se emplean 2 o 3 intervalos. El criterio para esta decisión es la distancia mínima total entre los elementos y su centroide. Si se configuran 2 intervalos, el límite entre intervalos se ajusta al valor más alto del primer grupo, mientras que, en el caso de 3 intervalos, se ajusta a los valores más altos del primer y segundo grupos.

Las diferentes tipologías resultantes del paso anterior aparecen acompañadas de información estadística sobre cada una: p. ej. intervalos de cada parámetro, número total

de edificios, número total de edificios incluidos y representatividad de la categoría en el conjunto del parque edificado. Cada tipología se representa con un color distinto.

También se utiliza una visualización en 3D de la ciudad para identificar qué edificios pertenecen a cada tipología, así como su distribución geográfica. Si la categorización no arroja unos resultados satisfactorios para el usuario, es posible retroceder a la fase anterior y modificar los criterios seleccionados.

Para el caso de la zona de estudio de Santiago de Compostela el proceso de categorización se ha llevado a cabo utilizando los siguientes parámetros e intervalos:

- Año de construcción: 2 intervalos. 1298 – 1800 y 1800 – 1986
- Uso principal: 1 intervalo. Solo residencial. Si bien el proceso automático de categorización se configura a todos los valores, a efectos de este caso de estudio solo se contemplaron los edificios residenciales.
- Número de fachadas: Todos los valores.
- Grado de protección: Todos los valores.

Los resultados de la configuración seleccionada para el caso de estudio se muestran en la Tabla 2-1. Se identificaron 32 tipologías para edificios residenciales en la zona de estudio. Solo 8 de ellas superaban el umbral de representatividad, fijado en el 3%. Las tipologías seleccionadas abarcaban un total de 605 edificios a considerar, lo que representaba el 86,55% del número total de edificios residenciales del área seleccionada (699 edificios residenciales).

Tabla 2-1 Identificación de las tipologías. [30]

IDENTIFICATION OF BUILDINGS TYPOLOGIES					SELECT REPRESENTATIVE TYPOLOGIES			
Use	Number of Facades	Year of Construction	Level of Protection	Typology Code	Number of Buildings	Total Buildings	Historical and Residential Buildings	Selected Typologies (3% Threshold)
Residential	1	<1800	4	1111	9	1,12%	1,29%	
			3	1112	2	0,25%	0,29%	
			2	1113	0	0,00%	0,00%	
			0	1114	1	0,12%	0,14%	
		>=1800	4	1121	73	9,07%	12,07%	5
			3	1122	7	0,87%	1,00%	
			2	1123	1	0,12%	0,14%	
			0	1124	7	0,87%	1,00%	
	2	<1800	4	1211	22	2,73%	3,64%	6
			3	1212	30	3,73%	4,96%	2
			2	1213	1	0,12%	0,14%	
			0	1214	0	0,00%	0,00%	
		>=1800	4	1221	264	32,80%	43,64%	7
			3	1222	85	10,56%	14,05%	3
			2	1223	3	0,37%	0,43%	
			0	1224	38	4,72%	6,28%	1
	3	<1800	4	1311	8	0,99%	1,14%	
			3	1312	10	1,24%	1,43%	
			2	1313	10	1,24%	1,43%	
			0	1314	1	0,12%	0,14%	
		>=1800	4	1321	60	7,45%	9,92%	8
			3	1322	33	4,10%	5,45%	4
			2	1323	4	0,50%	0,57%	
			0	1324	10	1,24%	1,43%	
4	<1800	4	1411	0	0,00%	0,00%		
		3	1412	3	0,37%	0,43%		
		2	1413	5	0,62%	0,72%		
		0	1414	0	0,00%	0,00%		
	>=1800	4	1421	7	0,87%	1,00%		
		3	1422	3	0,37%	0,43%		
		2	1423	2	0,25%	0,29%		
		0	1424	0	0,00%	0,00%		
Non residential				106	13,17%	15,16%		
						Selected Typologies	8	
						Selected Buildings	605	
						Representativity	86,55%	

En lo que a los edificios residenciales respecta, la mayor parte de ellos presentan únicamente 2 fachadas debido a la configuración urbana de la zona de estudio. También se puede apreciar que la mayoría de los edificios del casco antiguo han sido reconstruidos después del año 1800. El nivel de protección de los edificios residenciales es bajo (3 o 4). La tipología más común es la representada por edificios residenciales de 2 fachadas, reconstruidos a partir del año 1800 y con un nivel de protección 4, que es el más bajo.

En la Figura 2-3 se muestra la distribución geográfica de las tipologías en una visualización 3D de la herramienta de categorización. Hay que tener en cuenta que en esta categorización solo se contemplan los edificios residenciales, aunque el modelo de ciudad 3D incluye también otros no residenciales. También se muestra el conjunto completo de tipologías resultantes, junto con sus parámetros representativos e información estadística, tal y como aparece en la herramienta de categorización.



Figura 2-3 Distribución geográfica de las tipologías y los resultados de la categorización

La misma herramienta de categorización se ha adaptado para su aplicación a la mitigación de desastres relacionada con el cambio climático utilizando el modelo de Donostia.

Para el caso de la zona de estudio de Donostia – San Sebastián el proceso de categorización se ha llevado a cabo utilizando los siguientes parámetros e intervalos:

- Año de construcción: 2 intervalos. 1750 – 1923 y 1923 – 2014
- Uso principal: Todos los valores.
- Dispone de sótano: Todos los valores.
- Grado de protección: Todos los valores.

Se han identificado 7 tipologías que superan el umbral de representatividad, fijado en el 3%. Las tipologías seleccionadas abarcaban un total de 1825 edificios a considerar, de un total de 2262, lo que representa el 80,68% del número total de edificios.

La herramienta permite, además, obtener la vulnerabilidad y el riesgo de edificios en los que se dispone de más información semántica. Para ello se ha generado el modelo de los edificios cercanos al río Urumea ya que son los edificios más vulnerables ante estas situaciones.

Después de realizar la categorización de los edificios de Donostia, se ha realizado trabajo de campo para recoger datos de los edificios representativos de cada categoría. Después

se ha calculado la vulnerabilidad para cada edificio representativo y se ha extrapolado el resultado para toda la zona de estudio, dando el mismo valor a todos los edificios pertenecientes a la misma categoría (ver Figura 2-4).



Figura 2-4 Visualización de la vulnerabilidad de los edificios

La herramienta permite también combinar los resultados de la categorización con servicios de mapas con información adicional ofrecidos mediante servicios siguiendo el estándar *Web Map Service* (WMS). En la figura el ejemplo de mapa de inundabilidad de la zona de estudio de Donostia (ver Figura 2-5).

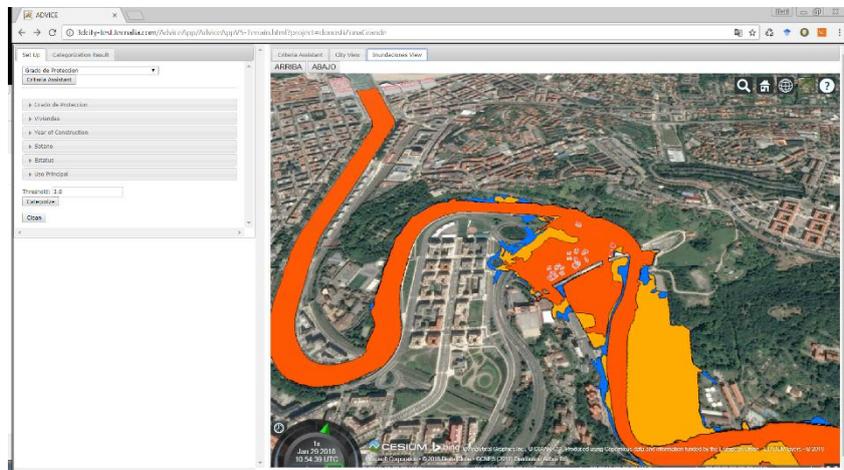


Figura 2-5 Visualización de mapas de inundaciones a través de servicios WMS

Utilizando esta herramienta de categorización que se basa en un modelo 3D urbano en CityGML ha sido posible identificar los edificios característicos de las diferentes categorías. Después, las categorías resultantes de la herramienta de categorización se analizan de forma individual a través de un edificio representativo de dicha tipología, para así identificar las soluciones de rehabilitación energética en el primer caso, y las soluciones de mitigación de inundabilidad en el segundo caso. De esta forma, es posible extrapolar los resultados a los edificios de dicha tipología y se puede medir el impacto de la intervención en cada tipología específica teniendo en cuenta la representatividad de dicha tipología en el distrito.

2.3.2 Plataforma colaborativa para la definición y seguimiento de un Plan Especial de Reforma Interior (PERI)

Contexto

Los Planes Especiales de Reforma Interior (PERI) tienen como objeto la mejora urbanística del espacio urbano: descongestión, creación de dotaciones, saneamiento, problemas de circulación y otros problemas análogos. El PERI permite definir una operación integrada de reforma interior, que intente copar todos los ámbitos y problemas del entorno urbano. La administración es quien desarrolla un PERI, y estos son los planes especiales con una mayor incidencia práctica en los entornos urbanos. El proceso de definición del PERI requiere una comunicación con los ciudadanos para recabar sus necesidades y problemas en una primera etapa y posteriormente para recoger el *feedback* a través de mejoras o sugerencias tras la comunicación de una propuesta inicial por parte de la administración. Adicionalmente las intervenciones asociadas a este plan incorporan ayudas públicas que hay que controlar y justificar minuciosamente para lo que es necesaria información veraz y precisa tanto de los propietarios como de los inmuebles. La información debe estar acompañada de acreditación identificativa y soporte gráfico. Para obtener dicha información es necesario llevar a cabo inspecciones técnicas y visuales. El proceso de generación del plan es largo y tedioso y requiere de la participación de diferentes agentes (administración, ciudadanos y profesionales).

Las nuevas tecnologías ofrecen soluciones cada vez más avanzadas de generación y recopilación de datos, con lo que se está generando un volumen de datos que crece significativamente y son datos de diferentes dominios, naturaleza y escala. Además, los avances tecnológicos están causando una reubicación y distribución de la información

que anteriormente se administraba principalmente de forma local. El concepto de la nube cada vez es más común y el trabajo colaborativo emerge en la gestión de información personal y profesional [31]. Las plataformas de colaboración en la nube ponen en valor el concepto de servicio de software frente al producto.

Se hace necesario identificar herramientas que faciliten la colaboración en la definición, gestión y seguimiento de los planes de regeneración urbana en general y los PERI en particular. El contexto actual es el idóneo para trasladar el paradigma de gestión colaborativa al ámbito de la gestión urbana, optimizando de forma notable los tiempos y costes, accesibilidad a datos, capacidad de tomar decisiones complejas, e incluir la participación ciudadana dentro del ciclo de decisiones, dando lugar a una verdadera gestión municipal digital. Para poder dar respuesta a estas exigencias las administraciones necesitan sistemas de información de la ciudad integrales, que garanticen la compatibilidad de las distintas aplicaciones que se desarrollen sobre ellos con las ya existentes y agilizando la gestión y mantenimiento de la información.

La definición de los modelos 3D urbanos, la creación y edición colaborativa de dichos modelos tiene varias ventajas en los procesos de gestión urbana; facilitando el trabajo sobre el modelo único siempre actualizado y la realización de trabajos en paralelo. La componente 3D facilita la representación de la información completa y fácilmente entendible por cualquier usuario. Estos modelos facilitan la cooperación de expertos en diferentes áreas, contribuyendo con sus conocimientos en la generación de un modelo único.

Modelo

El problema planteado arriba requiere de un modelo de representación de la información que permita integrar información heterogénea, un modelo que cubra múltiples dominios y que evolucione con el tiempo. Requiere además de una gran capacidad semántica, que permita un razonamiento sobre el modelo y evalúe las interacciones multi-dominio. La propuesta se basa en utilizar dos de los principales modelos de datos estándares a escala ciudad y edificio, CityGML e *Industry Foundation Classes* (IFC) respectivamente.

El modelo 3D urbano está cada vez más reconocido como un soporte adecuado a la hora de integrar, armonizar y almacenar la información de una ciudad y hacerla accesible a todos los interesados (ciudadanos, administradores de ciudades, empresas o

investigadores) [10]. Los modelos 3D urbanos van a permitir transformar la forma en que la ciudad enfrenta los desafíos sociales, económicos y ambientales.

El caso de estudio desarrollado para la puesta en práctica de la Plataforma colaborativa para la definición y seguimiento de un PERI tiene como escenario el Casco Histórico de Vitoria – Gasteiz. La elección de dicho entorno obedece en primer lugar a su tamaño y configuración, así como por la disponibilidad tanto de información gráfica como semántica existente sobre él (tales como cartografía vectorial, ortofotografías, modelo digital del terreno o datos correspondientes al PERI).

Se ha generado un modelo CityGML del casco histórico de Vitoria – Gasteiz con edificios en LOD0, LOD1 y LOD2. Ha sido necesario también extender el CityGML para incluir todos los datos temáticos necesarios. Los datos geométricos del modelo han sido procesados también para obtener nuevos parámetros semánticos (tales como área, orientación principal o medianeras, entre otros). Los modelos CityGML e IFC han sido desplegados en formato de fichero en un servidor generado para dicho propósito.

Un aspecto importante del modelo de este caso de estudio es que se ha ido completando a lo largo del tiempo. La representación simplificada de los edificios en CityGML se ha completado con la representación detallada del modelo IFC de un edificio. También es posible ampliar o modificar los datos semánticos de los edificios en los modelos CityGML e IFC, y se puede completar el modelo con nuevos recursos asociados a los elementos del modelo.

Aplicación

La plataforma colaborativa ha sido desarrollada en el marco del proyecto U3DCLOUD, cuyo objetivo es el desarrollo de una plataforma colaborativa basada en la nube para la creación y edición semiautomática de modelos 3D urbanos, dando soporte a la toma de decisiones complejas que impliquen varios agentes y ámbitos de decisión. Se trata de un dominio de gran relevancia y que sintetiza la complejidad multi-agente y multi-escala que se pretende acometer. La plataforma se basa en modelos de datos (CityGML e IFC) y servicios *Open Geospatial Consortium* (OGC) estándares. El modelo digital abarca desde la escala de la ciudad hasta los detalles del edificio, con una transición transparente para el usuario.

La plataforma se ha desarrollado sobre una arquitectura modular dividida en tres bloques principales (ver Figura 2-6):



Figura 2-6 Arquitectura de la plataforma U3DCloud

- **Sistema de Información Urbana 3D (SIU-3D)**, representa el modelo de información centralizada, basado en la filosofía *Common Data Environment* (CDE), que proporciona un único punto de acceso a la información, que está estructurado y vinculado adecuadamente, y se genera a partir de múltiples fuentes y conectado a diferentes herramientas. El SIU-3D permite la representación multiescala y se basa en modelos de datos estándares (CityGML e IFC).
- **Infraestructura de servicios en la nube**, proporciona acceso al SIU-3D a través de estándares abiertos (basados en *Representational State Transfer* (REST) y OGC). Esta capa está dividida en servicios de bajo nivel y de alto nivel. Entre los servicios de bajo nivel, destacan los servicios de carga, descarga, acceso y consulta de la información contenida en el modelo. Entre los servicios de alto nivel se encuentran los servicios de generación de modelos 3D, procesamiento geométrico, procesamiento semántico e interoperabilidad.

- **Las aplicaciones de usuario**, basadas en la infraestructura definida (modelo y servicios), se pueden desarrollar varias aplicaciones para generar y utilizar la información urbana. El prototipo desarrollado consta de tres aplicaciones:
 - **Aplicación de ámbito ciudad**: un plugin de ArcGIS que facilita la generación de modelos 3D urbanos en CityGML. Este plugin se basa en el servicio de generación de modelos 3D incluido en la capa de servicios, que proporciona algunos procesos automáticos para la creación semi-automática de la geometría de los modelos a partir de datos del catastro y LIDAR y la posterior semantización parcial del modelo.
 - **Aplicación U3DCloud-Web**: una aplicación web 3D que permite la visualización de ciudades y edificios con diferentes niveles de detalle. Incluye funcionalidades para visualización basada en capas y la posibilidad de subir y descargar recursos asociados a elementos del modelo (por ejemplo, modelos 3D detallados, imágenes o incidencias)
 - **Aplicación de ámbito edificio**: permite la edición y visualización del modelo 3D del edificio. Este modelo puede generarse con una herramienta externa basada en el estándar IFC, que después se sube al SIU-3D. Se ha desarrollado una aplicación basada en Unity para Tablet para visualizar, interactuar y actualizar el modelo en base a la infraestructura del servicio para mantener la integridad del modelo.

Basada en la plataforma arriba descrita, diferentes agentes participan en el proceso de definición, edición y seguimiento de un Plan Especial de Reforma Interior (PERI). El municipio es el encargado de coordinar el proceso, pero los ciudadanos y técnicos de diferentes dominios también pueden compartir información y participar en el proceso. El proceso basado en la plataforma desarrollada se divide en 4 etapas. Cada una de estas etapas se divide en varias actividades en las que colaboran diferentes agentes. Para llevar a cabo las actividades de cada participante, se utilizan diferentes herramientas en diferentes dispositivos, los cuales comparten el modelo de información, y una infraestructura de servicio única, que permite mantener la integridad del modelo y la interoperabilidad entre los componentes de la plataforma. A continuación, se describen algunos detalles del proceso en cada una de las 4 etapas (ver Figura 2-7). Se detallada el proceso que incluye las etapas, los agentes y las herramientas. Las diferentes

herramientas, servicios o interfaces en los que se basa el flujo de información se representan con diferentes colores.

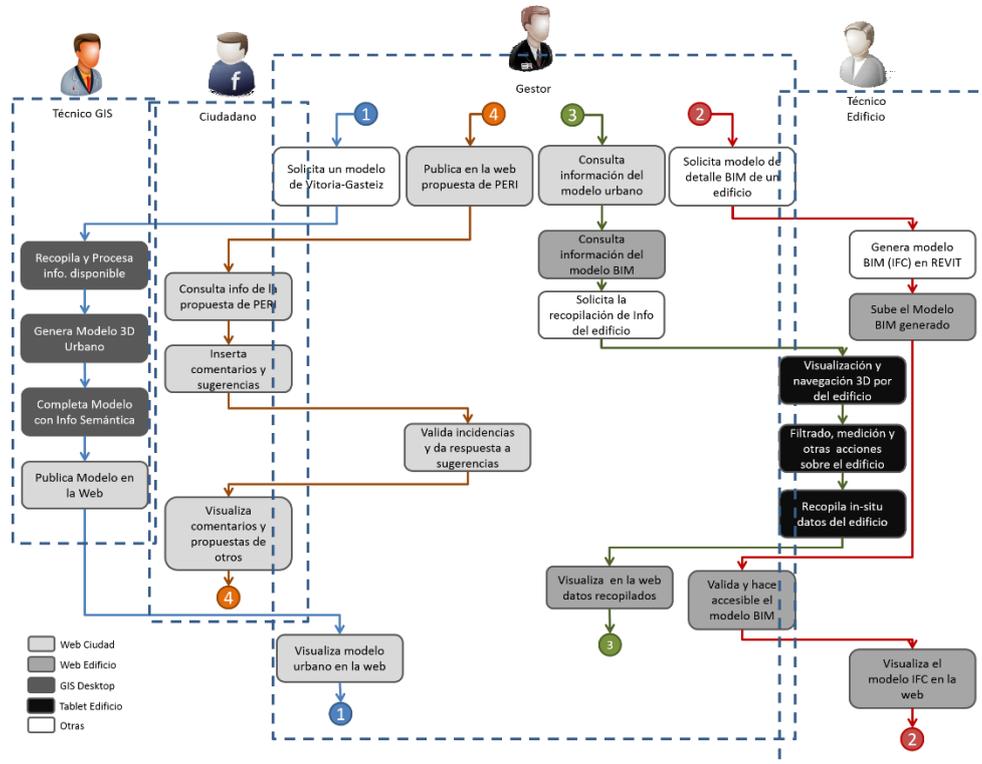


Figura 2-7 *Workflow* de plataforma colaborativa basada en servicios para la definición y seguimiento de un PERI

(1) Generación del modelo 3D urbano

El gestor del distrito solicita un modelo 3D del centro urbano de Vitoria-Gasteiz. A continuación, el técnico *Geographic Information System* (GIS) recopila y procesa información disponible en ArcGIS, para generar el modelo en LOD2 mediante *plugins* desarrollados en el proyecto, empezando con la generación geométrica y posteriormente completando la semántica del modelo CityGML. Una vez generado el modelo, éste se publica en la Web. Mediante la aplicación web del proyecto es posible visualizar el modelo generado, así como navegar y hacer diferentes consultas sobre la información contenida en el modelo. La Figura 2-8 muestra el plugin desarrollado en ArcGIS para la generación del modelo.



Figura 2-8 Plugin desarrollado en ArcGIS para la generación del modelo

(2) Incluir información detallada de edificios

El gestor solicita un modelo en detalle (IFC) para un edificio concreto. El técnico del edificio genera un modelo IFC utilizando la herramienta REVIT para dar respuesta al gestor. El modelo generado se sube a través de la aplicación web a la plataforma. El modelo pasa la correspondiente validación por parte del gestor y una vez validado el recurso aparece visible en la aplicación web. La Figura 2-9 muestra el edificio modelado en REVIT y el mismo edificio visualizado a través de la aplicación web del proyecto.

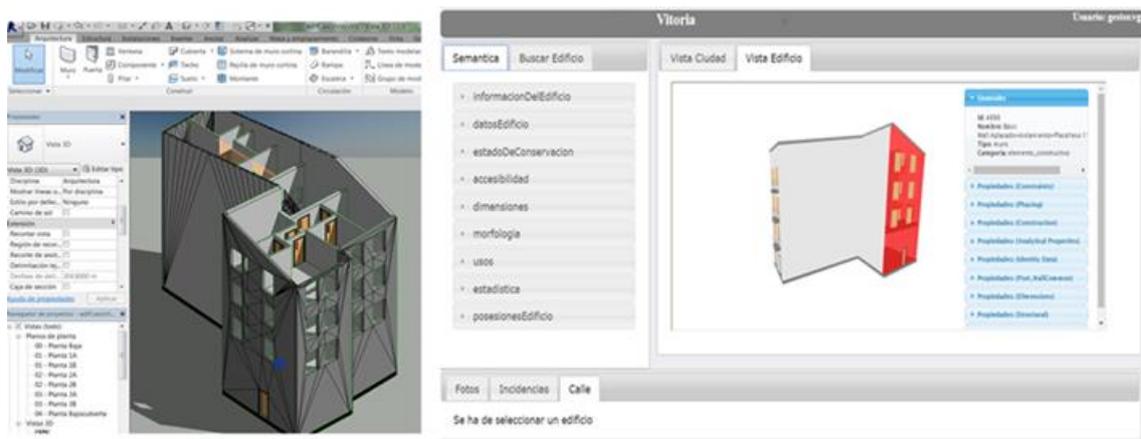


Figura 2-9 Generación de un modelo BIM en Revit / Visualización del modelo IFC en el visor 3D

(3) Diagnóstico de la situación actual del área de estudio

El gestor analiza la información del modelo urbano desde la aplicación web (tanto a escala distrito como edificio a edificio). Posteriormente, el mismo solicita al técnico completar la información que no se incluye en el modelo de detalle del edificio. El técnico se desplaza al edificio y utiliza la aplicación móvil de ámbito de edificio para visualizar el modelo 3D, lo analiza, toma anotaciones y medidas y modifica o completa los datos de modelo. Las modificaciones se actualizan en la plataforma. Los datos recopilados por el técnico durante la inspección se visualizan por parte del gestor desde la aplicación web. La Figura 2-10 muestra la apariencia de la aplicación para tablet de ámbito de edificio.

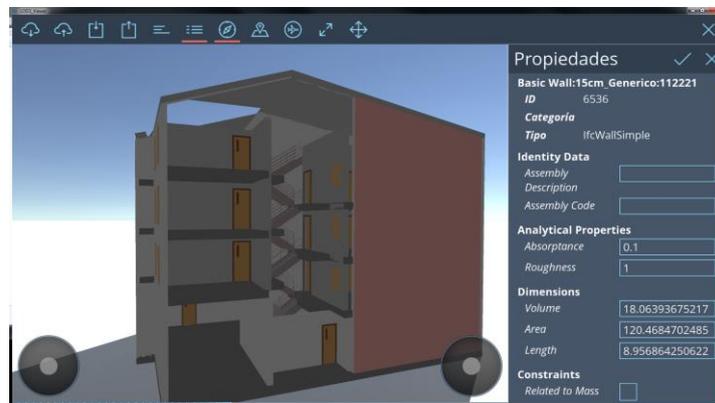


Figura 2-10 Recopilación de datos in-situ del edificio utilizando la tablet

(4) Participación ciudadana para la definición del plan de regeneración

Una vez recogida toda la información relevante, el gestor define y publica en la web una propuesta de PERI a compartir con la ciudadanía. Los ciudadanos consultan la información disponible sobre los inmuebles de la zona de estudio a través de la aplicación web. Además de la información asociada a cada inmueble el plan recogerá información de recursos asociados a los elementos del entorno (edificios u otros). Los ciudadanos pueden incluir sus comentarios, incidencias o recursos fotográficos para completar el PERI, enviar *feedback* o proponer modificaciones. Los recursos introducidos por los ciudadanos deben ser revisados y validados por el gestor como paso previo a la publicación de los mismos. La Figura 2-11 muestra un ejemplo de presentación de recursos fotográficos asociados a un elemento del entorno sobre el modelo de referencia para la definición del PERI.

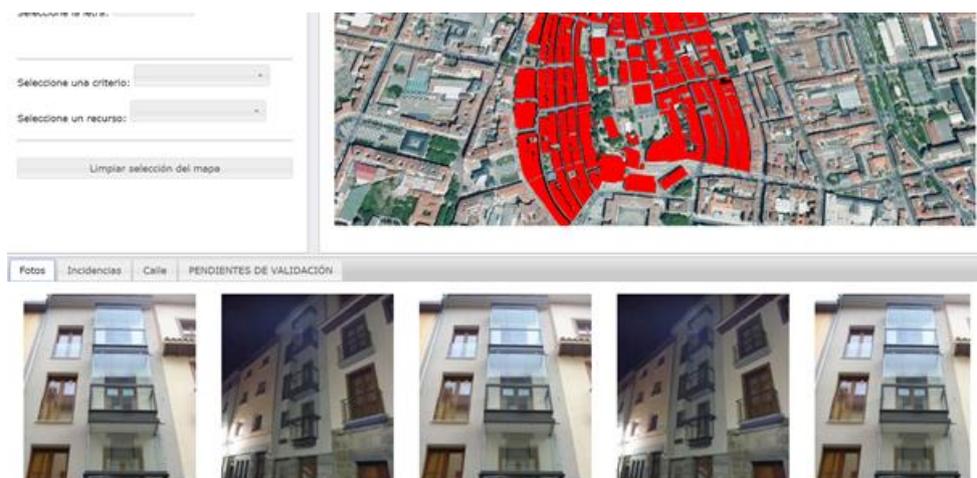


Figura 2-11 Visualización de imágenes del edificio seleccionado

2.3.3 Evaluación del impacto de aplicación de medidas de conservación de energía

Contexto

El interés en mejorar la eficiencia energética y el confort térmico de los entornos urbanos históricos surge de una doble demanda: la necesidad sociocultural de preservar las ciudades históricas y la necesidad ambiental de reducir la demanda energética global de los edificios. El patrimonio cultural de una ciudad ha demostrado ser una característica importante para el bienestar de los ciudadanos y los edificios históricos son muy apreciados [32] [33], pero los habitantes de las ciudades con frecuencia eligen edificios más modernos ya que se perciben como más cómodos que los históricos. Los edificios que no se usan rara vez se conservan, lo que convierte el abandono de las ciudades históricas en un importante problema de conservación urbana. El Consejo de Europa, en la Declaración de Ámsterdam de 1975, promovió el concepto de conservación integrada que establece la mejora de la habitabilidad y la calidad de vida de sus ciudadanos como uno de los principales objetivos de la conservación urbana [34]. Existe una estrecha relación entre la habitabilidad y la eficiencia energética.

La mejora del comportamiento energético de los edificios históricos no solo es un tema cultural, sino también importante en términos de objetivos medioambientales globales en Europa, ya que más del 40% de las viviendas europeas se construyeron antes de 1960 [35]. Como concluye una investigación reciente, "faltan protocolos específicos destinados a proporcionar soluciones bien equilibradas para la mejora de la eficiencia energética en edificios históricos e históricos" [36]. Esta falta es aún más notable si se aborda el problema en la escala urbana.

La selección de estrategias de rehabilitación a escala urbana ofrece beneficios significativos, pero hace que el proceso sea largo y costoso. Por lo tanto, se necesitan métodos o herramientas para establecer una evaluación rápida que facilite la toma de decisiones estratégicas y un análisis más profundo de un número reducido de alternativas. Del mismo modo es necesario evaluar el impacto de la aplicación de las medidas de conservación para la mejora energética a escala urbana.

Modelo

Los estudios a escala urbana pueden consumir grandes cantidades de energía y dinero debido a la cantidad de información que se requiere, a menudo como resultado del trabajo de campo. Los modelos 3D son una solución cada vez más aceptada para almacenar y mostrar información tanto en la construcción como en las escalas urbanas. Ofrecen los mismos beneficios que 2D GIS, pero proporcionan una mayor funcionalidad a través de la tercera dimensión [37].

Al igual que en el caso de estudio 1, se ha generado un modelo CityGML de Santiago de Compostela y se ha completado con información semántica disponible de fuentes de datos públicas. Después, el modelo CityGML ha sido enriquecido con atributos adicionales necesarios para la aplicabilidad del modelo a la metodología de evaluación. Esta metodología permite la toma de decisiones a la hora de identificar la idoneidad de la aplicación de medidas de conservación energéticas en distritos históricos en etapas tempranas. La implementación del modelo urbano requiere:

- (1) La definición de la jerarquía de los elementos clave según los elementos definidos en el esquema CityGML. La estructura de los elementos de la ciudad que se considera bajo el esquema CityGML es genérica debido a su universalidad. Como el concepto de distrito no está definido en CityGML, se ha definido utilizando el concepto de agrupación `cityObjectGroup`, que permite la agregación de objetos de ciudad de acuerdo con los criterios definidos por el usuario. Cada edificio puede contener instalaciones, que en este caso se enfocan en las instalaciones energéticas. También se representan las superficies del edificio en detalle, tales como paredes y tejados y también las puertas y ventanas que contienen.
- (2) La extensión del modelo CityGML. Se diseñaron dos extensiones de dominio de aplicación (*Application Domain Extension* - ADE) CityGML específicas en el proyecto EFFESUS para estructurar toda la información semántica necesaria para

la toma de decisiones y gestión de medidas de conservación de energía (*Energy Conservation Measures - ECM*) en áreas urbanas.

Extensión de dominio de patrimonio cultural

El ADE de Patrimonio Cultural incluye información a escala distrito, el edificio y la envolvente del edificio. El modelo estructura la información con respecto a la importancia histórica según tres aspectos principales (visual, físico y espacial) y para elementos a diferentes escalas: la escala distrito, escala del edificio (tales como muros externos, tejados, ventanas o balcones) y nivel de vivienda (tales como paredes internas, puertas o acabados interiores). La información sobre los principales materiales de construcción y sus propiedades se establece al nivel de la envolvente del edificio, y la consideración general de la integridad y el estado de conservación se establece a nivel del edificio (por ejemplo, tipo histórico, integridad o estado físico). El grado de protección y las restricciones legales para la preservación del patrimonio se identifican a nivel de edificio. Una descripción detallada del ADE de patrimonio cultural puede encontrarse en el Capítulo 3 de la tesis de A. Egusquiza [30]. El esquema *Unified Modeling Language* (UML) del ADE de patrimonio cultural se muestra en la Figura 2-12.

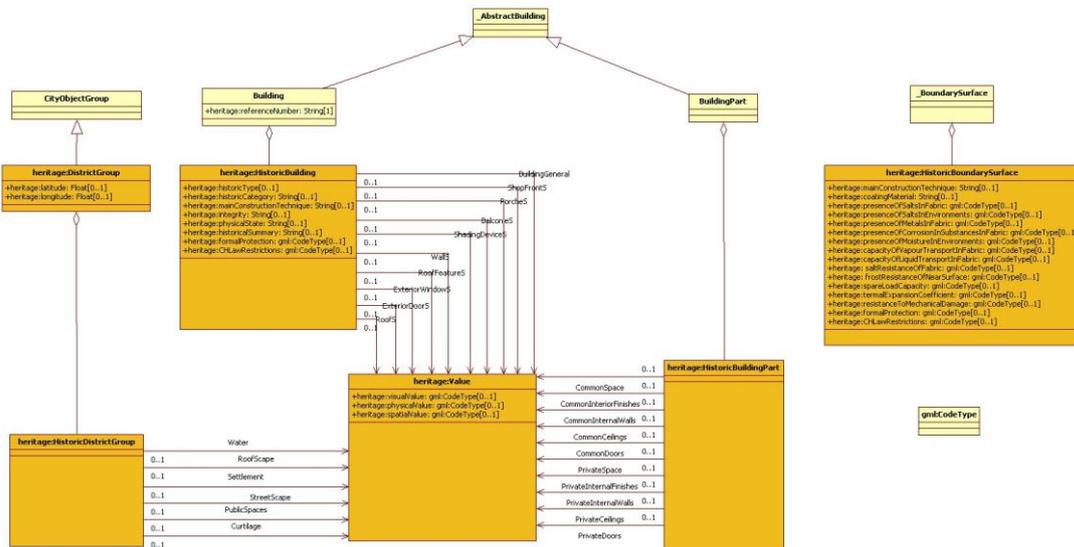


Figura 2-12 Diagrama UML para el ADE de Patrimonio Cultural

Extensión de dominio de comportamiento energético

La ADE de comportamiento energético incluye información a diferentes escalas: distrito, edificio, envolvente del edificio e instalaciones del edificio (instalaciones de demanda y generación). La información relacionada con el clima se referencia a nivel de distrito. En el nivel del edificio, se establece la información relacionada con la geometría (es decir, superficie del suelo y altura de los pisos), ocupación (ganancias internas, masa térmica e infiltración de aire) y uso (temperatura de confort y estrategia de ventilación). Las propiedades del material (valores U y valor G), el tipo (fachada o medianera) y el tamaño de las ventanas y la relación entre las áreas opacas y de apertura se identifican en el nivel de envolvente del edificio. En el nivel de instalación del edificio, se hace referencia a las instalaciones de energía (tipo y eficiencia). Una descripción detallada del ADE de comportamiento energético se puede encontrar en el capítulo 3 de la tesis de A. Egusquiza [30]. El esquema UML del ADE de comportamiento energético se muestra en la Figura 2-13:

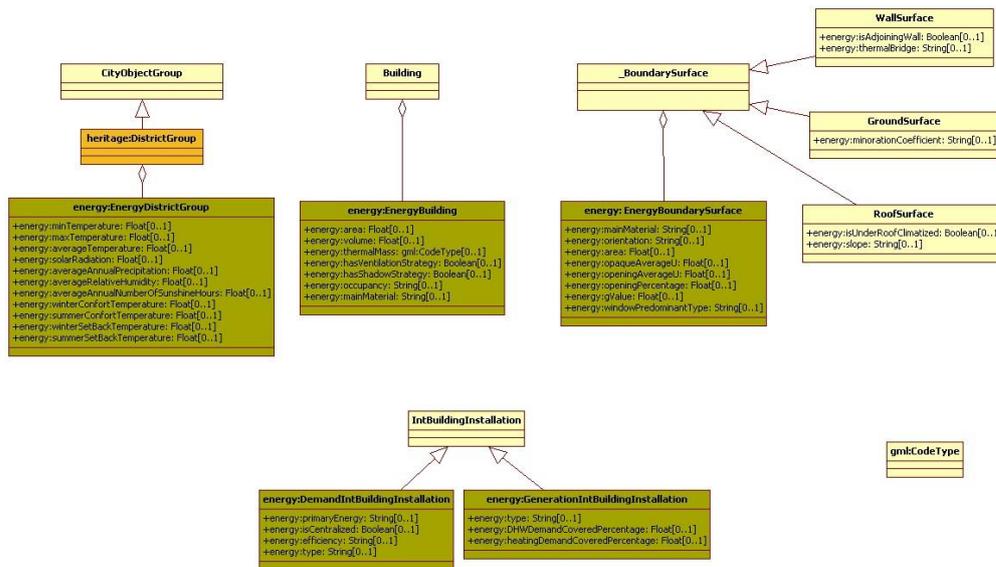


Figura 2-13 Diagrama UML para el ADE de Comportamiento Energético

A la hora de semantizar el modelo la mayoría de los parámetros a nivel de edificio se han recopilado del catastro español y se han procesado de manera que se incluyan automáticamente en el modelo CityGML. Por otro lado, la mayoría de los parámetros a nivel de distrito se obtienen de la base de datos climática de la Agencia Estatal de Meteorológica (AEMET) y se ha completado manualmente en el modelo CityGML.

Como resultado, todos los edificios (819) del distrito están representados en LOD2 con fachadas y tejados independientes, y la altura del edificio se ha obtenido a partir de datos LIDAR. Los datos geométricos del modelo han sido procesados también para obtener nuevos parámetros semánticos (área, orientación principal, medianeras, etc.).

En este caso de estudio el modelo CityGML resultante ha sido desplegado en una base de datos para después poder utilizar servicios estándares de acceso tales como el *Web Feature Service*.

Aplicación

El objetivo de esta herramienta es desarrollar un método que permita evaluar en etapas tempranas el impacto de la aplicación de medidas de conservación de energía (*energy conservation measures* (ECM)) a escala urbana utilizando fuentes de datos abiertos o públicas. Para ello, se utiliza un modelo basado en CityGML para estructurar la información necesaria para la evaluación de su aplicabilidad y los impactos positivos y negativos de la aplicación a nivel urbano de cada ECM. La idoneidad de las medidas de conservación de energía se puede evaluar equilibrando el impacto positivo con la preservación de los valores culturales (mejora de la eficiencia energética y el confort térmico) con sus valores negativos (impacto sobre la autenticidad e integridad de los elementos del edificio).

Este trabajo se ha llevado a cabo dentro del proyecto EFFESUS y para su validación se ha seleccionado el centro histórico de la ciudad de Santiago de Compostela. Se han seleccionado 11 ECM para ser probados en la zona seleccionada de Santiago de Compostela. El criterio principal para la selección de estos 11 ECM ha sido elegir soluciones con un claro impacto en el comportamiento energético y en la materialidad de los edificios. Se han seleccionado soluciones que mejoran tanto la hermeticidad del edificio como las características térmicas de la envolvente.

El método desarrollado proporciona una evaluación rápida de los ECM que requiere poco o ningún trabajo de campo y proporciona resultados a escala urbana, pero al mismo tiempo las medidas deben aplicarse en la escala de edificio, teniendo en cuenta las características geométricas de los edificios y sus elementos principales (fachada, tejados, ventanas e instalaciones) y la aplicabilidad de las medidas según las características o requisitos de conservación.

En estos casos, un modelo de información a múltiples escalas que combina de manera coherente información geométrica y semántica es la opción más adecuada para la representación de la información. En este caso se ha utilizado un modelo urbano basado en el estándar CityGML. En el marco del proyecto EFFESUS, se ha desarrollado un modelo urbano a varias escalas que integra información sobre patrimonio cultural y energía.

El comportamiento energético del edificio se ha calculado considerando la norma internacional *International Organization for Standardization (ISO) 13790: 2008* ("Rendimiento energético de los edificios - Cálculo del uso de energía para calefacción y refrigeración de espacios"), basado en un método mensual cuasi estático. Cada edificio ha sido tratado como una única zona, con uso residencial y valores mensuales de datos climáticos típicos. Se desarrolló una herramienta específica para automatizar los cálculos de la evaluación del impacto del patrimonio a nivel de edificio para todos los edificios en el caso de estudio seleccionado. La herramienta se basa en la implementación de la norma citada y una implementación ad hoc para la evaluación de la aplicación de las ECM.

La aplicación del método al área de estudio de Santiago de Compostela arroja resultados numéricos que permiten comparar diferentes estrategias de reacondicionamiento aplicables al área urbana (ver Figura 2-14).

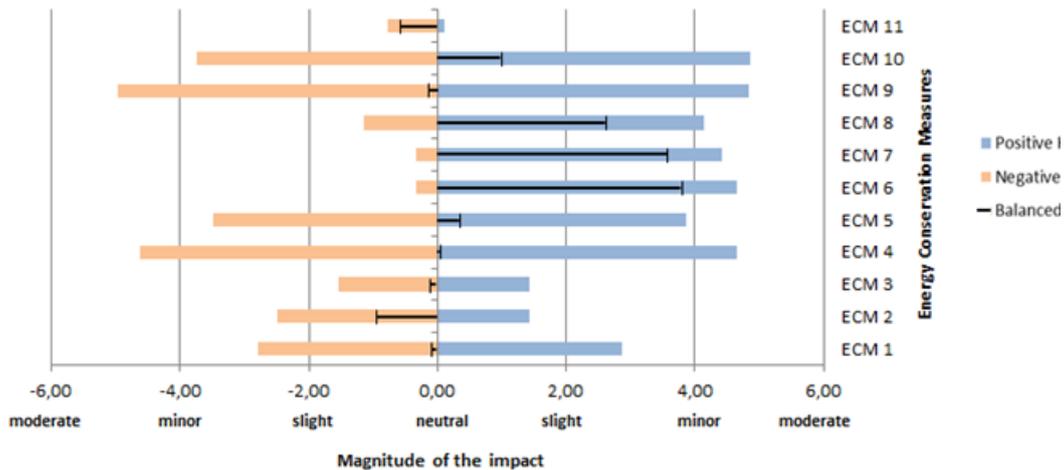


Figura 2-14 Comparación de los resultados de los diferentes ECM

De la misma forma, es posible visualizar el resultado en el modelo CityGML mediante un visor 3D web basado en la librería gráfica Cesium (ver Figura 2-15 y Figura 2-16).

En la Figura 2-15 se presenta el impacto negativo de un ECM en concreto para todo el distrito histórico de Santiago de Compostela. La escala de color, que va desde el amarillo hasta el naranja oscuro, representa cada vez un impacto más negativo.

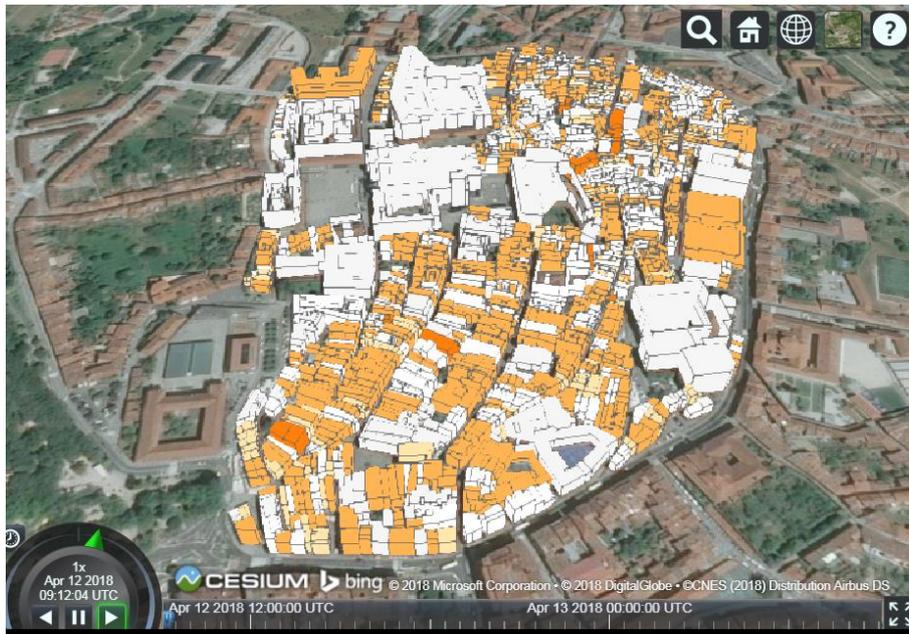


Figura 2-15 Visualización del impacto negativo de los ECM

En la Figura 2-16 se presenta el impacto positivo de un ECM en concreto para todo el distrito histórico de Santiago de Compostela. La escala de color, que va desde el azul claro hasta el azul oscuro, representa cada vez un impacto más positivo.

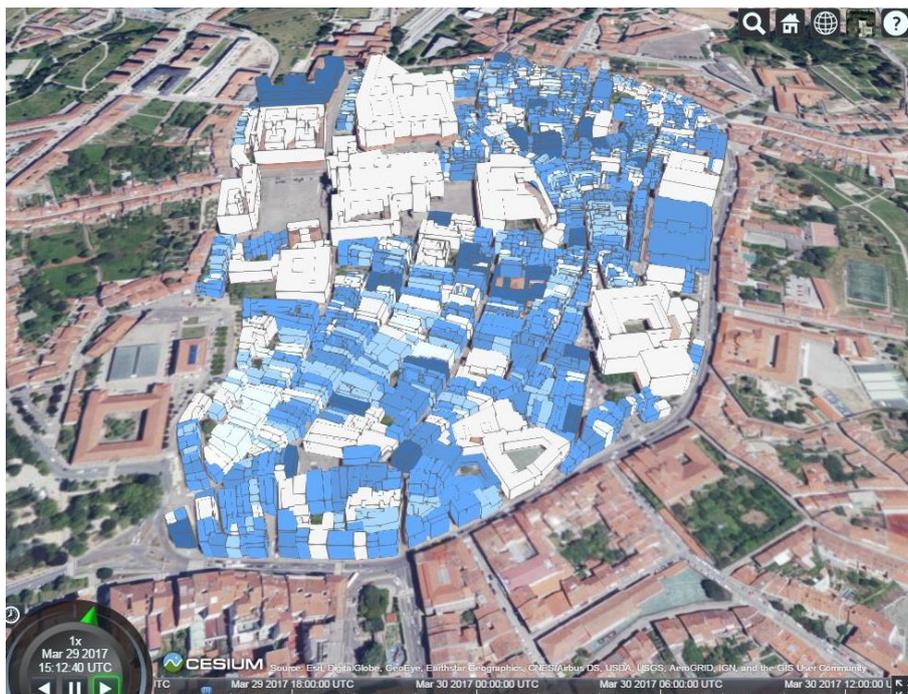


Figura 2-16 Visualización del impacto positivo de los ECM

2.4 Análisis de los casos de estudio

Una vez descritos y analizados los casos de estudio presentados en el apartado anterior es posible identificar requisitos para los modelos 3D urbanos. Algunos de los requisitos son comunes a los casos de estudio y otros específicos de uno o algunos de ellos. Para ello se ha seguido el método de investigación con estudio de casos, que permite validar los requisitos aplicándolos en casos de estudio reales [15].

2.4.1 Comparación de los casos de estudio

En los tres casos de estudio se ha generado un modelo CityGML como paso previo al desarrollo de las aplicaciones. Para el caso de estudio (CE) 1 se ha generado el modelo CityGML del casco histórico de Santiago de Compostela, incluyendo 819 edificios. Para este CE se han generado también otros elementos de la ciudad, tales como carreteras o zonas verdes. El CE2 tiene como zona de interés el casco histórico de Vitoria – Gasteiz, el cual incluye 709 edificios. El CE3 tiene la misma zona de estudio que el CE1, sin embargo, la generación es diferente ya que los datos necesarios en las extensiones son diferentes de un caso a otro, y en el CE3 no se han generado otros elementos de la ciudad. En todos los CE la generación de los edificios se ha realizado en LOD0, LOD1 y LOD2.

Esta generación conlleva mucho tiempo en: la obtención de datos, el preprocesado de los datos y la generación del modelo CityGML. Además, el proceso seguido para la generación de los modelos de los casos de estudio ha sido diferente en cada caso. En algunos casos se utilizan datos del catastro de España, en otros casos el catastro autonómico e incluso OSM, etc.

Uno de los objetivos durante la generación del modelo CityGML en todos los CE ha sido intentar minimizar el trabajo manual, intentando automatizar con diferentes procesos algunas de las tareas durante la generación.

Ha sido necesario también extender el modelo en todos los CE. Para ello, primero ha sido necesario analizar los requisitos de las aplicaciones e identificar qué nuevos parámetros y objetos urbanos son necesarios. Después, se han creado cada una de las extensiones en CityGML.

Después, ha sido necesario semantizar el modelo en todos los casos de estudio. En algunos casos la semantización se realiza directamente utilizando la información disponible en las fuentes de datos utilizadas en la generación (como el catastro), de cara a facilitar (y evitar el trabajo manual) a la hora de completar la información del modelo. Mientras que en otros (CE1) ha sido necesario también realizar trabajo de campo para una zona específica en la que se necesitan datos detallados.

Otra forma de completar la información semántica ha sido procesando la geometría del modelo (para obtener el área, orientación principal o medianeras) u obteniendo nuevos datos mediante inferencia tales como periodo de construcción o material de construcción, entre otros. Estos procesos han sido necesarios en los tres CE.

Una peculiaridad del CE2 ha sido que el modelo 3D urbano ha ido creciendo a lo largo del tiempo. Es decir, un técnico puede subir a la plataforma el modelo BIM de un edificio en concreto; o bien ampliar los datos temáticos de un edificio en concreto en CityGML en cualquier momento. Esto ha dado como resultado un modelo 3D urbano más completo a lo largo del tiempo.

En el CE2 es posible también que la ciudadanía participe completando el modelo añadiendo recursos como imágenes o incidencias. Si bien en este caso se ha realizado mediante una aplicación (con lo cual se minimizan las posibilidades de crear errores en el modelo), puede darse el caso que el completado se realice de otras formas, lo que puede

llevar a tener errores en el modelo. En el CE1 ha sido necesario seguir completando el modelo CityGML una vez creado. Ya que después de realizar la categorización de los edificios se ha completado la información en detalle de los edificios representativos. Por ello, cada vez que un usuario realizaba algún cambio en el modelo CityGML se ha comprobado que el modelo sigue siendo válido en todos los sentidos.

Por eso, al generar los modelos 3D urbanos en CityGML de los tres CE o cada vez que se ha realizado un cambio, se han utilizado diferentes herramientas (de terceros y desarrollos propios) para validar el modelo CityGML. Primero se ha comprobado si el CityGML es válido, también se ha validado que la geometría en *Geography Markup Language* (GML) estaba correctamente representado. Asimismo, ha sido necesario comprobar en un visor 3D si la geolocalización, altitud de los edificios sobre el *Digital Terrain Model* (DTM) y apariencia del modelo es la correcta. Como es de suponer, este proceso es repetitivo, costoso y aun así no se garantiza que el modelo sea totalmente válido. Del mismo modo, cada cambio implicaba también (además de la validación), tener que desplegar el modelo en el servidor de forma manual y comprobar si todo se había realizado correctamente.

En los casos de estudio 1 y 3 el modelo CityGML resultante ha sido desplegado en una base de datos, mientras que en el CE2 el modelo se ha desplegado en formato fichero.

En el CE3 se ha calculado el comportamiento energético del edificio considerando la norma internacional ISO 13790: 2008. El proceso se ha automatizado para calcular los resultados de la aplicación de las diferentes variantes (cada ECM) a cada uno de los edificios.

2.4.2 Requisitos de los modelos 3D urbanos

En este apartado se identifican una serie de requisitos para los modelos 3D urbanos a partir del análisis de los Casos de Estudio (CE) que se ha realizado. En este trabajo de investigación se ha utilizado CityGML como lenguaje para expresar el modelo 3D urbano. Los requisitos se presentan estructurados por categorías que representan cada una de las principales etapas del ciclo de vida de un modelo 3D urbano. Adicionalmente se incluyen requisitos generales asociados a varias etapas o independientes de las mismas. La Tabla 2-2 describe el requisito y los casos de estudio en los que se ha detectado dicho requisito.

Tabla 2-2 Requisitos de los modelos 3D urbanos

	Descripción	CE1	CE2	CE3
Generales	El modelo debe permitir representar información de diferentes elementos de la ciudad (edificios, zonas verdes, carreteras, mobiliario urbano, etc.)	X		
	El modelo debe permitir la representación de los elementos urbanos con diferentes niveles de detalle	X	X	X
	El modelo debe permitir la interoperabilidad entre diferentes herramientas de modelado, simulación, etc. El uso de modelos de datos estándares facilita la interoperabilidad.			X
	El modelo debe permitir la reutilización de la información en él contenida para diferentes aplicaciones, separando la información geométrica de las propiedades semánticas	X	X	X
	El modelo debe permitir una representación coherente de la geometría y semántica de los elementos del modelo.	X	X	X
	El modelo debe ser fácilmente extensible tanto en extensión espacial del modelo (añadir un nuevo edificio, barrio, etc.) como en ámbito de la información (añadir información de un nuevo dominio)		X	
	El modelo debe ser generado de forma semi-automática minimizando el trabajo manual	X	X	X
Generación geométrica	El modelo debe permitir representar la planta 2D de los edificios (LOD0 de CityGML)	X	X	X
	El modelo debe permitir representar los edificios como cubos 3D (LOD1 de CityGML)	X	X	X
	El modelo debe permitir representar los edificios como cubos 3D distinguiendo entre fachadas y tejados (LOD2 de CityGML)	X	X	X
	El modelo debe permitir representar la información correctamente georreferenciada	X	X	X

	El modelo debe permitir la representación de información del DTM y las ortofotografías de la zona	X	X	X
	El modelo debe estar correctamente posicionado en altitud respecto al DTM	X	X	X
Generación semántica	El modelo debe permitir completar la información semántica en cualquier momento	X	X	X
	El modelo debe incluir nuevos parámetros semánticos que no estén representados en el <i>core</i> del modelo	X	X	X
	El modelo debe permitir incorporar información procedente de diferentes dominios, tales como energética, económica, espacial, patrimonial e incluso impacto ambiental.	X	X	X
	El modelo debe permitir obtener nuevos parámetros del procesado de la geometría del modelo, tales como: área, volumen, orientación principal o medianeras.	X	X	X
Mantenimiento	El modelo debe permitir editar la información geométrica	X	X	X
	El modelo debe permitir editar la información semántica	X	X	X
	El modelo debe permitir la semantización masiva de forma semi-automática		X	
Validación	El modelo debe estar correctamente estructurado (que sea un XML válido y cumpla el esquema XSD)	X	X	X
	El modelo debe ser geoméricamente válido (no tener puntos consecutivo-repetidos, el primer y último punto de un polígono deben ser iguales, etc.)	X	X	X
	El modelo debe representar la geometría de los edificios correctamente referenciada respecto a la ortofotografía de la zona	X	X	X
Despliegue	El modelo debe poder ser desplegado en una base de datos	X		X
	El modelo debe poder ser desplegado como fichero en un servidor		X	

Se han identificado también otros requisitos que, aunque están ligados a estos casos de estudio, han sido requisitos de otros proyectos (FASUDIR⁷, HOLISTEEC⁸) y casos de estudio en los que se ha trabajado.

- El modelo debe contener información heterogénea georreferenciada, tales como sensores o puntos de interés.
- El modelo debe permitir la agrupación de edificios y otros elementos urbanos en distritos o áreas definidas.
- El modelo debe permitir añadir texturas de los edificios.
- El modelo debe almacenar información de la geometría de los principales elementos constructivos de los edificios (puertas, ventanas, fachadas y tejados)

Estos requisitos se traducen en la definición de un modelo de datos común, multiescala, genérico, interoperable y que contiene la información semántica y geométrica necesaria para la gestión, la toma de decisiones para la mejora de la sostenibilidad y su posterior gestión y mantenimiento.

2.4.3 Soporte para los requisitos identificados en las herramientas existentes

Las compañías líderes en el sector de las aplicaciones de diseño y modelado de la industria de la construcción e ingeniería (*Architectural, Engineering and Construction - AEC*) están expandiendo lentamente la escala tecnológica dando el salto a soluciones que abordan la escala urbana. Este es sin duda un reto que deberán abordar en el futuro más cercano si quieren seguir manteniendo su posición en el mercado. Entre dichas soluciones cabe destacar:

- **Autodesk InfraWorks:**⁹ Autodesk InfraWorks ofrece las últimas tecnologías de modelado 3D, visualización y colaboración basadas en la nube. Consiste en un software de diseño preliminar que ayuda a ver un proyecto en 3D y mejorar los resultados al combinar y conectar datos e información de diversos archivos de trabajo y programas, para tener una mejor visualización del proyecto, hacer mejores análisis, compartir los resultados y tomar mejores decisiones. InfraWorks

⁷ <http://fasudir.eu/>

⁸ <http://www.holisteecproject.eu/>

⁹ <http://www.autodesk.com/products/infracore-360/overview>

es la iniciativa de Autodesk para aplicar los procesos BIM a la escala urbana a través del diseño y gestión de las infraestructuras.

- **Bentley, 3D Cities¹⁰**: La solución Ciudades 3D proporciona un modelo de información por encima y por debajo del suelo de la ciudad que comprende el entorno construido y la infraestructura de servicios públicos. Se trata de una solución para administrar, editar, analizar y visualizar grandes modelos de ciudades en 3D para aplicaciones prácticas, como mapas temáticos avanzados, estudios de línea de visión, simulaciones de ruido, planificación urbana, y el catastro 3D.

Existen actualmente múltiples herramientas diseñadas específicamente para el modelado de información urbana en 3D. Algunos ejemplos de estas soluciones son:

- **Ciudades 3D de Google**: El ejemplo más conocido de Ciudades en 3D es el programa de Google para desarrollar el modelo 3D de la ciudad a partir de las aportaciones de los usuarios y visualizar directamente dicho modelo sobre Google Earth. Esta aproximación permite generar un modelo geométrico en 3D de una ciudad completa a partir de aportación de usuarios y apuesta de las administraciones locales. Sin embargo, se trata únicamente de un modelo geométrico sin información semántica asociada a dicha geometría. El formato de archivo utilizado para representar las ciudades 3D de Google es *Keyhole Markup Language* (KML).
- **CityEngine**. Otra aproximación a la generación de ciudades en 3D es la propuesta por la herramienta CityEngine, recientemente adquirida por Esri. Esta herramienta se basa en el modelado procedural de una ciudad en 3D con alto grado de detalle a partir de información 2D y reglas o parámetros configurables por el usuario que ofrecen una respuesta visual inmediata. Esta herramienta se centra en la representación visual de una ciudad genérica obtenida a partir de parámetros. Esta opción resulta muy útil para crear ciudades virtuales en 3D para incorporar en videojuegos y otras aplicaciones, donde el grado de fidelidad con que representa la realidad no es máximo. Estas ciudades carecen también de información semántica.

¹⁰ <https://www.bentley.com/en/solutions/3d-cities>

- **VirtualcityPUBLISHER¹¹**: Es una solución desarrollada para crear y publicar aplicaciones geoespaciales interactivas en 3D a partir de datos basados en CityGML. De forma rápida e intuitiva es posible mostrar en un navegador web datos geoespaciales en 3D combinados con los servicios de datos geoespaciales 2D existentes (e.g. WMS, WFS). Esta solución se basa en otros servicios y soluciones desarrollados por el mismo proveedor (e.g. virtualcityMAP, virtualcityDATABASE) para componer una de las suites de aplicaciones sobre modelos urbanos 3D disponibles en la actualidad.
- **CiberCity 3D¹²**: Ha puesto en marcha una plataforma de cartografía 3D de Ciudades inteligentes para ayudar a las autoridades municipales y los planificadores a visualizar sus pueblos y ciudades en 3D, incluyendo los futuros planes de la ciudad existente y, desarrollos y esquemas propuestos.
- **CityPlanner¹³**: Es una solución ofrecida por la compañía Agency 9 para la planificación urbana en 3D. Ofrece características avanzadas de visualización urbana y exploración de los elementos urbanos. También ofrece la posibilidad de realizar análisis de sombreado y líneas de visión en tiempo real. Todo ello sobre un entorno 100% web fácil de usar, actualizar y compartir.
- **GeoBrowser3D¹⁴**: Ofrece una solución de visualización geoespacial en 3D ideal para la toma de decisiones en entornos urbanos. Esta solución permite introducir modelos e información de servicios geoespaciales existentes para analizar y tomar decisiones. Todo ello sobre el propio navegador web y con técnicas de visualización avanzada.
- **SmarterBetterCities¹⁵**: Ofrece una plataforma on-line llamada CloudCities que es un producto que permite alojar, compartir y visualizar modelos de ciudades en 3D. Esta plataforma permite crear y publicar modelos en 3D a través de múltiples plataformas de forma eficaz, desde aplicaciones desktop web, dispositivos móviles o gafas de realidad virtual. También ofrece la posibilidad de visualizar resultados y capas de información conectada al modelo 3D urbano.

¹¹ <http://www.virtualcitysystems.de/en/products/virtualcitypublisher>

¹² <http://www.cybercity3d.com/>

¹³ http://agency9.com/#section_products

¹⁴ <http://geobrowser3d.com/>

¹⁵ <http://www.smarterbettercities.ch/cloudcities/>

- **Cityzenith**¹⁶: 5D Smart City ayuda a las ciudades a conectar, visualizar y gestionar sus datos por medio de una simulación 3D interactiva, intuitiva y en tiempo real, con la que todo el mundo, desde el alcalde hasta el ingeniero sito en el campo, puede usar de forma sencilla. Los usuarios apuntan, hacen clic, tocan y deslizan para conseguir la información que necesitan en tiempo real — datos abiertos, datos de dispositivo IoT y datos de medios sociales — sin tener formación o con muy poca.

La mayoría de las herramientas descritas se centran en la generación geométrica del modelo, situando en un segundo plano la generación de información semántica. Además, se centran principalmente únicamente en los edificios, sin tener en cuenta otros elementos de la ciudad. Del mismo modo, aunque estas herramientas permiten generar y visualizar el modelo generado, generan el modelo en un momento en concreto y no dan mecanismos para el mantenimiento del mismo.

Ninguna de las herramientas anteriormente descritas aborda adecuadamente todos los requisitos anteriormente descritos. Por un lado, las herramientas descritas no permiten crear extensiones del modelo durante la generación, no pudiendo añadir información temática adicional. Por otro lado, la validación del modelo es un requisito que se deja de lado por la mayoría de las herramientas. De hecho, algunos de los problemas (p.e. disponer de un modelo que pueda ser desplegado) parecen no haber sido considerados en absoluto.

2.5 Conclusiones

Como ha podido verse en este capítulo, la implementación del concepto *smart city* requiere del desarrollo de servicios que faciliten la gestión y planificación urbana. En los casos de estudio descritos en la sección anterior se han presentado algunos ejemplos que, basándose en un modelo 3D urbano, permiten ofrecer alguno de estos servicios.

Estos casos de estudio han permitido aprovechar el potencial del modelo 3D urbano en herramientas con diferentes finalidades: categorizar el parque edificado de una ciudad, realizar la gestión colaborativa para la definición y seguimiento de un PERI y evaluar el impacto de aplicación de ECM. Los casos de estudio son un ejemplo que reflejan la

¹⁶ <http://www.cityzenith.com/>

utilidad de los modelos 3D urbanos. La necesidad de disponer de estos modelos va en aumento al igual que su utilización en diferentes aplicaciones [11].

A partir de los casos de estudio, se han identificado una serie de requisitos que deben cumplir los modelos 3D urbanos. Se han analizado también herramientas existentes para crear, editar, publicar y visualizar modelos 3D urbanos e identificado si cumplen los requisitos previamente definidos.

La identificación de requisitos y conclusiones a partir de los casos de estudio analizados ha permitido identificar líneas de trabajo e investigación a partir del análisis del trabajo realizado utilizando modelos 3D urbanos. Los casos de estudio representan una muestra representativa de los servicios para la gestión y planificación de ciudades descritos en la sección 2.2.

Para estos casos de estudio los modelos 3D han sido generados sin seguir ninguna metodología. Sin embargo, se han identificado procesos similares y repetitivos durante la generación del modelo en los tres casos. Este proceso hay que mejorarlo y al ser un proceso (la generación de modelos 3D) cada vez más frecuente, conviene crear procesos genéricos para poder aplicarlo más veces. Por eso, el capítulo 3 de esta tesis profundiza sobre la necesidad de crear y actualizar modelos 3D urbanos de manera semiautomática. Para ello se describe un flujo de trabajo para la generación y se describen también ciertas operaciones de mantenimiento que después son implementadas y validadas en una aplicación.

De la misma forma, durante el mantenimiento de los modelos 3D urbanos de los casos de estudio se ha trabajado de forma colaborativa. Es decir, cualquiera de los agentes o *stakeholders* participantes en el proceso podría realizar cambios en el modelo 3D urbano y notificar (mediante email) los cambios realizados en el mismo. Uno de los principales problemas es que se han tenido que validar cada vez que se ha realizado algún cambio. Esta validación se ha realizado de forma semi manual y sin utilizar ninguna metodología. Además, es un proceso repetitivo, que necesitaba bastante tiempo y siempre se encontraban errores más adelante. Por todo esto, el capítulo 4 de esta tesis presenta una solución para facilitar el mantenimiento regular de unos modelos de ciudad 3D en CityGML. La solución se fundamenta en la estrategia y las herramientas de despliegue continuo empleadas en el desarrollo de *software* y se adapta al problema derivado de crear, mantener y desplegar modelos CityGML validados.

3 Metodología de generación y actualización semiautomática de modelos CityGML

3.1 Introducción

Un modelo 3D urbano georreferenciado es una solución para el almacenamiento y despliegue de datos urbanos que está ganando adeptos, siendo el estándar CityGML una buena opción para ello. Combina información 3D y semántica en un único modelo de datos, con lo que facilita el uso y la interoperabilidad entre modelos [38]. Este tipo de modelos de datos sirve para representar, gestionar y mantener datos urbanos para diversas aplicaciones, como pueden ser la gestión de desastres, la planificación urbana y del tráfico, la seguridad, las telecomunicaciones, la navegación o el turismo [11].

Actualmente la mayoría de las herramientas de generación de modelos 3D urbanos se centran principalmente en los edificios, sin tener en cuenta el resto de elementos de la ciudad. Además, la actualización tanto de la geometría como de la semántica de los modelos 3D urbanos sigue siendo un reto hoy en día.

En este capítulo se ha propuesto un flujo de trabajo para la creación de modelos 3D urbanos (generación y mantenimiento) a partir de fuentes de datos generalmente gratuitos y de naturaleza heterogénea como solución para poder crear modelos 3D urbanos de forma semiautomática con datos tanto geométricos como semánticos. Este flujo de trabajo engloba el diseño del modelo de datos, el volcado de información geométrica y semántica (incluida la derivación automática de propiedades semánticas) y la a menudo olvidada fase de mantenimiento.

El capítulo está estructurado como se describe a continuación: tras revisar otros trabajos anteriores en el apartado Trabajo relacionado, se describe el flujo de trabajo para la creación y el mantenimiento de modelos de ciudad 3D en Ciclo de vida de ciudades en 3D. Después se detalla cómo se han automatizado algunos pasos del *workflow* en Implementación. A continuación, se describe una aplicación real del *workflow* en Aplicación. El capítulo se cierra con las conclusiones.

3.2 Trabajo relacionado

3.2.1 Modelos de datos para ciudades en 3D

Un modelo de datos es una forma de representación de información que define los datos, la estructura de dichos datos (su tipología, la forma en que se relacionan, etc.), las operaciones que pueden realizarse con ellos (añadir, eliminar, modificar, recuperar) y las restricciones referentes a su integridad (aquellas condiciones que todos los datos deben respetar para que se transmita la información de manera correcta). Una concepción más amplia nos permite entender al modelo de datos como aquel que sirve para describir los componentes que actúan en una cierta realidad y la manera en que se vinculan entre sí.

Mediante la información geométrica se puede representar un edificio en 3D, sin embargo, sólo aporta información visual. Con la información semántica, es posible enriquecer el modelo de datos con datos tales como: número de plantas del edificio, materiales utilizados, la función del edificio o quién es el propietario, entre otros. Con lo cual es necesario disponer de un modelo de datos que sea capaz de almacenar simultáneamente información 3D y semántica. Para almacenar información de edificios es necesario unir conceptos desde el ámbito de un distrito hasta el de un edificio, es decir, unir conceptos GIS y BIM en un mismo modelo de datos.

La Tabla 3-1 resume la comparación entre diferentes modelos de datos de intercambio de datos 3D [39]:

Tabla 3-1 Comparación entre diferentes modelos de datos de intercambio 3D [39]

	DXF	SHP	VRML	X3D	KML	Collada	IFC	CityGML	3D PDF
Geometry	++	+	++	++	+	++	++	+	++
Topology	-	-	0	0	-	+	+	+	-
Texture	-	0	++	++	0	++	-	+	+
LOD	-	-	+	+	-	-	-	+	-
Objects	0	+	+	+	-	-	+	+	+
Semantic	+	+	0	0	0	0	++	++	+
Attributes	-	+	0	0	0	-	+	+	+
XML based	-	-	-	+	-	-	+	+	-
Web	-	-	+	++	++	+	-	+	0
Georef.	+	+	-	+	+	-	-	+	+
Acceptance	++	++	++	0	++	+	0	+	++

- not supported; 0 basic; + supported; ++ extended

Como se ve en esta tabla, CityGML es el modelo de datos más completo, siendo el único que admite diferentes LOD y uno de los más completos en cuanto a la inclusión de información semántica y geométrica. El uso generalizado de CityGML en toda Europa es otra de sus ventajas, por ejemplo la mayoría de las ciudades alemanas tienen un modelo CityGML al menos en su nivel más bajo de detalle [40].

Se han analizado con más profundidad algunos de los modelos de datos de intercambio de datos con más potencial para representar modelos de ciudades en 3D. A continuación, se presentan algunas de las principales características de los modelos de datos analizados.

Green Building XML (gbXML)

El esquema Green Building XML (gbXML)¹⁷, fue desarrollado para facilitar la transferencia de información de edificios almacenados en sistemas CAD. Este esquema permite intercambiar datos entre herramientas *Computer-Aided Design* (CAD) y herramientas de análisis de eficiencia energética, lo que permite mejorar la interoperabilidad entre distintas herramientas y simplifica la transferencia de información entre diferentes agentes. gbXML tiene el apoyo de los proveedores de sistemas CAD

¹⁷ <http://www.gbxml.org/>

principales como Autodesk, Graphisoft, y Bentley. A raíz del desarrollo de la funcionalidad para importar / exportar sobre las herramientas de modelado más importantes, gbXML se ha convertido en un esquema estándar de facto de la industria.

Como el objetivo de gbXML es facilitar la transferencia de información de edificio tiene una relación cercana a la especificación *Industry Foundation Classes* (IFC). Comparando gbXML e IFC [41], se llega a la conclusión de que gbXML es una especificación más sencilla y fácil de entender. Además, añade información sobre eficiencia energética, es posible extenderlo fácilmente y es interoperable entre distintas herramientas.

Geography Markup Language (GML)

Geography Markup Language (GML) [42] es un estándar internacional para el intercambio de datos espaciales definido por la OGC y la International *Organization for Standardization* (ISO) TC211. GML sirve como un lenguaje de modelado de sistemas geográficos, así como un formato de intercambio abierto para transacciones de información geográfica a través de la red.

GML contiene un amplio conjunto de primitivas que se utilizan para crear esquemas de aplicación específica o dominios de la aplicación. Estas primitivas son: características geométricas, sistema de coordenadas de referencia, topología, tiempo, características dinámicas, unidad de medida, direcciones y observaciones.

City Geography Markup Language (CityGML)

CityGML es un modelo de datos abierto en formato basado en XML para el almacenamiento e intercambio de modelos virtuales de ciudades en 3D [38]. Se trata de un esquema de aplicación para la versión 3.1.1 del GML3.

El objetivo del desarrollo de CityGML es llegar a una definición común de las entidades básicas, atributos y relaciones de un modelo de ciudad en 3D. Esto es especialmente importante ya que permite la reutilización de los mismos datos en diferentes campos de aplicación.

CityGML no sólo representa el aspecto gráfico de los modelos de ciudad, también representa las propiedades semánticas y temáticas, taxonomías y agregaciones. Incluye además un modelo geométrico y un modelo temático. El modelo geométrico permite definir homogénea y coherentemente las propiedades geométricas y topológicas de los

objetos espaciales del modelo 3D de la ciudad. Mientras que el modelo temático de CityGML emplea el modelo de la geometría de las diferentes áreas temáticas tales como: modelos digitales del terreno, los edificios, vegetación, cuerpos de agua, instalaciones de transporte o mobiliario urbano, entre otros. Otros objetos que no están explícitamente modelados pueden representarse utilizando el concepto de objetos genéricos y atributos y pueden realizarse extensiones del modelo de datos CityGML utilizando el *Application Domain Extensions* (ADE).

Virtual Reality Modeling Language (VRML)

Virtual Reality Modeling Language [43] fue diseñado como formato de archivo para crear objetos y mundos tridimensionales interactivos. Fue la primera tecnología reconocida como estándar ISO para crear, representar y transmitir elementos 3D a través de la Web mediante los navegadores.

Las ventajas principales de VRML son que es extensible, ya que puede ampliarse de una forma sencilla, el visualizador donde se represente es independiente de la plataforma, puede trabajar de forma eficaz con conexiones lentas y es posible interactuar con el mundo virtual.

COLLABorative Design Activity (COLLADA)

COLLABorative Design Activity [44] es un esquema XML diseñado para facilitar compartir y reutilizar recursos digitales 3D entre diferentes herramientas de contenidos digitales. Permite la especificación de una forma comprensible de geometría, efectos y *shaders*, física, animación y cinemática. El esquema es extensible para permitir añadir información específica para un programa o que no está soportado en el modelos de datos.

El objetivo de Collada es fortalecer y permitir el desarrollo de un sistema de distribución de contenidos más avanzado estandarizando una representación común intermedia. Está pensado para archivos de intercambio y no como estructura final. El archivo está diseñado para permitir la mayor compatibilidad posible entre distintas herramientas, así como ser comprensible.

Keyhole Markup Language (KML)

Keyhole Markup Language (KML) [45], es un formato basado en XML. Permite la creación de modelos y el almacenamiento de funciones geográficas como puntos, líneas,

imágenes, polígonos y modelos que es posible visualizar Google Earth, Google Maps y otras aplicaciones.

Google Earth procesa los archivos KML de una manera similar a como los navegadores web procesan los archivos HyperText Markup Language (HTML) y XML. Al igual que los archivos HTML, los KML cuentan con una estructura basada en etiquetas con nombres y atributos utilizados para poder visualizarlos, por lo tanto, Google Earth actúa como un navegador de archivos KML.

KML y GML tienen una estructura parecida, pero tienen diferentes propósitos. GML está definido para describir objetos geográficos complejos, pero no está diseñado para explotarlos (p.ej. visualizarlos) eficientemente. Mientras que KML define objetos geográficos básicos, como puntos, líneas y polígonos junto con su estilo y representación gráfica.

Architecture Engineering and Construction XML (aecXML)

Architecture Engineering and Construction XML [46] es un modelo de datos para Arquitectura, Ingeniería y Construcción. El objetivo del esquema aecXML consiste en ayudar a las empresas de software, empresas de construcción, instituciones académicas, fabricantes de productos, etc. a la hora de intercambiar datos de los procesos. Sirve para facilitar las comunicaciones relacionadas con el diseño, especificación, estimación, abastecimiento, instalación y mantenimiento de productos y materiales de construcción a través de Internet. El objetivo de aecXML es eliminar las barreras actuales para el intercambio de información de AEC.

Building Information Model XML (BIMXML)

Building Information Model XML¹⁸ describe los datos de la construcción como: sitios, edificios, plantas, espacios, equipos, atributos, etc. de modelos de edificios simplificados.

Este esquema XML se desarrolló como una alternativa a modelos a escala real de IFC para simplificar el intercambio de datos entre diversas aplicaciones AEC y para conectar los BIM a través de Servicios Web.

¹⁸ <http://bimxml.org/>

Industry Foundation Classes XML (ifcXML)

Industry Foundation Classes XML (ifcXML) [47] consiste en utilizar e implementar el estándar IFC utilizando la tecnología XML. El contenido de un archivo ifcXML se ve limitado por el estándar XML, la versión actual del esquema ifcXML y por los requisitos del esquema del IFC. El objetivo consiste en crear una representación reconocible y coherente que se puede prever en el momento de lectura y debe lograrse al escribir modelos ifcXML.

Comparativa

El objetivo de esta sección consiste en identificar un modelo 3D urbano que mejor se ajuste a los requisitos identificados en el capítulo anterior. Todos los modelos descritos están en XML, lo que facilita la validación e intercambio entre aplicaciones. gbXML, ifcXML, GML, CityGML y KML son modelos de datos reconocidos por instituciones de prestigio con lo que se mejora la interoperabilidad entre diferentes aplicaciones.

GML, CityGML y KML son los únicos que representan información del ámbito de un distrito/ ciudad (GIS). El resto se centran en los edificios, en los procesos relacionados a la hora de diseñar un nuevo edificio y las relaciones entre los diferentes agentes (BIM). CityGML también permite representar información sobre edificios, pero sin tanto detalle como los anteriores.

El inconveniente de GML y KML es que pueden almacenar geometría (con lo que se podría almacenar la geometría de un edificio) pero no están diseñados para almacenar información semántica de los mismos. CityGML, en cambio, ha sido diseñado para almacenar los dos tipos de información.

Con CityGML es posible almacenar información tridimensional a diferentes niveles de detalle, puede almacenarse información semántica, podemos almacenar información desde el ámbito de un distrito (GIS) hasta el de un edificio (BIM); toda la información se almacena en un único modelo de datos; es posible extenderlo para almacenar información de dominio no contemplada en el *core* y es un estándar internacional definido por la OGC.

Una vez analizados y comprados los diferentes modelos de datos; se ha elegido CityGML como el modelo datos para la representación de ciudades en 3D en esta investigación. En la siguiente sección se describe CityGML con más detalle.

3.2.2 CityGML

CityGML es el estándar más utilizado para la representación de modelos 3D urbanos. Este estándar definido por la OGC permite combinar información geométrica y semántica, soporta varios niveles de detalle [48] y hace posible entrelazar datos a diferentes escalas [49], desde el territorio en general hasta los edificios en particular [11] [50].

El objetivo del desarrollo de CityGML es llegar a una definición común de las entidades básicas, atributos y relaciones de un modelo 3D urbano. Lo que es especialmente importante, ya que permite la reutilización de los mismos datos en diferentes campos de aplicación.

CityGML no sólo representa el aspecto gráfico de los modelos de ciudad, también representa las propiedades semánticas y temáticas, taxonomías y agregaciones. CityGML incluye un modelo geométrico y un modelo temático. El modelo geométrico permite definir homogénea y coherentemente las propiedades geométricas y topológicas de los objetos espaciales del modelo 3D urbano.

El modelo temático de CityGML emplea el modelo de la geometría de las diferentes áreas temáticas tales como: modelos digitales del terreno, los edificios, la vegetación, cuerpos de agua, instalaciones de transporte o mobiliario urbano, entre otros. Otros objetos, que no están explícitamente modelados, pueden representarse utilizando el concepto de objetos genéricos y atributos.

CityGML garantiza un modelado coherente entre geometría y semántica. Aunque el estándar engloba algunas propiedades semánticas para los objetos definidos en CityGML, los datos presentes en el núcleo de CityGML no suelen ser suficientes para cumplir los requisitos del usuario o las aplicaciones a desarrollar en base al modelo urbano [51]. CityGML creó para ello el concepto de las extensiones de dominio de aplicación (ADE). Distintos grupos de investigación han definido varias ADE para diferentes dominios y finalidades [52][53], pero ninguna se ha integrado aún en un flujo de trabajo concreto para la generación y el mantenimiento de modelos 3D urbanos.

Entre las principales características de CityGML cabe destacar las siguientes:

Modelado multiescala

CityGML admite diferentes niveles de detalle (*Level of Detail* - LOD). Los LOD son necesarios para reflejar la recolección de datos de procesos independientes con diferentes requisitos de aplicación. Además, los LOD facilitan la visualización y análisis de los datos. En un conjunto de datos CityGML, el mismo objeto puede ser representado en diferentes LOD simultáneamente. Esto permite analizar y visualizar el mismo objeto con diferentes grados de resolución (ver Figura 3-1). A continuación, se describe cada LOD en detalle: LOD0: Es el modelo digital del terreno en dos dimensiones y medio. Un mapa o una imagen aérea pueden representarse mediante este nivel. LOD1: Es el modelo de bloques que comprende edificios mediante prismas con techos planos. LOD2: Diferencia las estructuras del techo y superficies de forma temática. LOD3: Denota los modelos arquitectónicos detallando paredes, tejados, balcones, etc. Pueden asignarse texturas de alta resolución a estas estructuras. Además, la vegetación y los objetos de transporte pueden representarse en este nivel. LOD4: Completa el LOD3 añadiendo estructuras interiores. Por ejemplo, los edificios pueden representarse con habitaciones, puertas, escaleras y muebles.

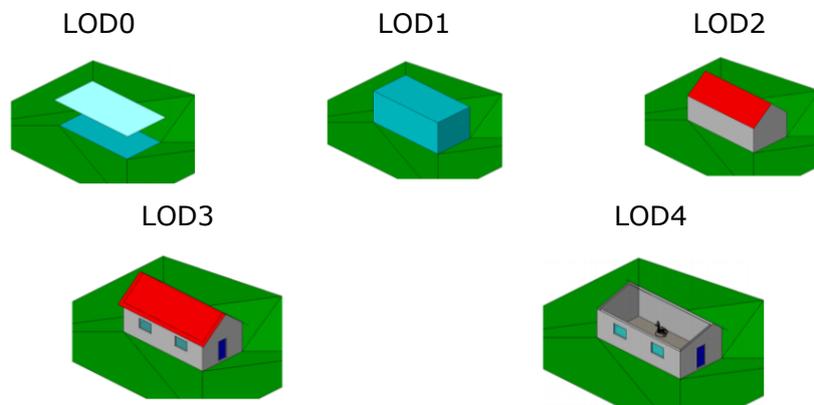


Figura 3-1 Los diferentes niveles de detalle en CityGML

Modelado coherente semántica y geoméricamente

Uno de los principios de diseño más importante de CityGML es el modelado coherente de la semántica y las propiedades geométricas/topológicas. En el nivel semántico, las entidades del mundo real se representan mediante características, tales como edificios, paredes, ventanas o habitaciones. La descripción también incluye los atributos, las relaciones y jerarquías de agregación entre las características. De este modo, parte de las relaciones entre características puede ser obtenido del nivel semántico sin contar con la geometría. Sin embargo, a nivel espacial, los objetos geométricos son asignados a las características representando su ubicación espacial y la extensión. El modelo consta de dos jerarquías: la semántica y geométrica en el cual los objetos están unidos por relaciones (ver Figura 3-2).

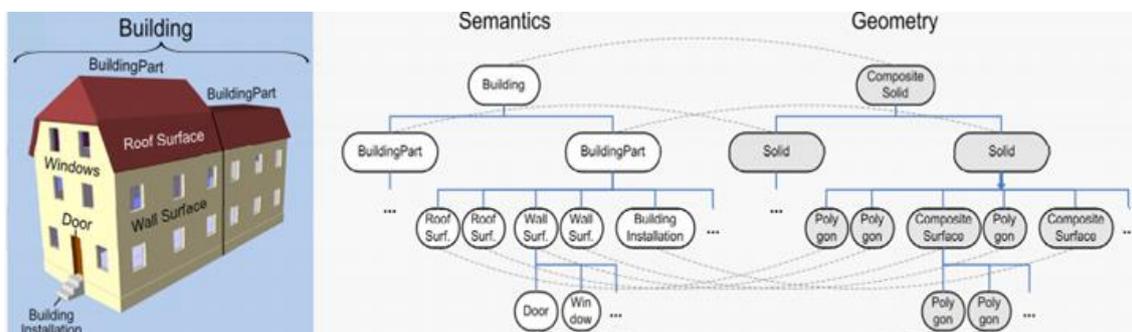


Figura 3-2 Modelado coherente semántica y geoméricamente en CityGML

Modularización

En el nivel temático CityGML define clases y relaciones para los objetos topográficos más relevantes de las ciudades y de los modelos regionales que abarcan: las estructuras, la elevación, la vegetación, cuerpos de agua, mobiliario urbano, etc. Además de la geometría y del contenido de la apariencia estos componentes temáticos permiten emplear los modelos virtuales de la ciudad 3D para las tareas de análisis en diversos dominios del uso como: simulaciones, minería de datos urbana, gestión de instalaciones e investigaciones temáticas.

Modelo extensible

CityGML está siendo diseñado como un modelo universal de información topográfica que define los tipos de objetos y atributos que son útiles para una amplia gama de aplicaciones. Sin embargo, es probable que en aplicaciones prácticas los objetos dentro del modelo 3D urbano necesiten algún atributo que no está definido explícitamente en

CityGML. Por otra parte, puede haber objetos en 3D que no están cubiertos por las clases temáticas de CityGML, para ello CityGML proporciona dos conceptos diferentes:

- **Objetos genéricos y atributos:** Este concepto permite la extensión de las aplicaciones CityGML en tiempo de ejecución. Cualquier elemento urbano puede ampliarse mediante atributos adicionales, cuyos nombres, tipos de datos y valores pueden ser proporcionados por una aplicación sin ningún cambio en el esquema de CityGML. Del mismo modo, las características que no están representadas en las temáticas predefinidas de CityGML pueden ser modeladas utilizando los objetos genéricos.
- **Application Domain Extensión (ADE):** Las extensiones del dominio de aplicación especifican los nuevos objetos y parámetros añadidos al modelo de datos CityGML. Por ejemplo, el número de habitantes de un edificio o la definición de los nuevos tipos de objeto. La diferencia entre las ADE y los objetos genéricos y atributos consiste en que un ADE tiene que ser definido en un nuevo archivo en el que se define el esquema XML con su propio *namespace*. Este archivo tiene que importar explícitamente la definición del esquema de los módulos extendidos de CityGML. La ventaja de este acercamiento es que la extensión está especificada formalmente y los documentos extendidos pueden ser validados contra CityGML y el esquema de ADE.

3.2.3 Proyectos e iniciativas relevantes

Existen actualmente un gran número de proyectos e iniciativas internacionales relacionadas con los modelos de datos urbanos en general y con el uso de CityGML en particular. Algunas de las más relevantes se presentan brevemente a continuación.

INSPIRE

En 2007 se aprobó, a nivel Europeo, la directiva INSPIRE [8] cuyo objetivo es crear un marco jurídico que permita establecer una infraestructura de datos espaciales [54] a nivel Europeo. La directiva INSPIRE permitirá disponer de más datos espaciales y que sean más fiables y poner al alcance de todos (administraciones, empresas y ciudadanos) dichos datos. Los problemas actuales de los datos espaciales son que: falta información, los datos no están debidamente documentados, no existe compatibilidad entre los datos o servicios

ofrecidos, ya que cada país tiene sus normas técnicas y todo esto impide la reutilización y el poder poner en común los datos.

Se ha realizado también una reflexión sobre considerar el 3D dentro de esta directiva. Múltiples países disponen de modelos de edificios 3D de ciudades, pero el inconveniente es que, normalmente, sólo están disponibles para grandes ciudades.

Dutch 3D pilot¹⁹

El Dutch 3D pilot se desarrolló entre marzo de 2010 y junio de 2011 y participaron en el mismo más de 65 organizaciones privadas, públicas y científicas con el objetivo de impulsar desarrollos 3D en los Países Bajos. Identificaron la necesidad de impulsar un estándar 3D en los Países Bajos. Para este propósito, se definieron casos de uso y se ejecutaron una serie de pruebas. Finalmente, el marco de estandarización 2D holandesa fue estudiada para permitir una extensión en 3D, siempre alineado con el estándar internacional CityGML.

German initiative InGeoForum²⁰

InGeoForum es el foro de información y cooperación compartida por los proveedores y usuarios de datos espaciales, sistemas y servicios de información geográfica. El objetivo principal de este foro es promover la información geográfica a nivel nacional y regional y fomentar el desarrollo de una infraestructura nacional y regional de datos espaciales.

Los modelos 3D urbanos y sus aplicaciones es un tema de gran actualidad para los gestores públicos. Se trata en primer lugar de identificar oportunidades significativas para los titulares de los datos y las organizaciones interesadas en el procesamiento y el uso final (ciudadanos, agentes públicos, empresas privadas). Este foro establece discusiones periódicas a través de mesas redondas para identificar áreas de aplicación e integración de los datos en 3D urbanos, requisitos de calidad y estructura de dichos datos en 3D, así como las posibles cadenas de valor añadido que se pueden definir desde los proveedores de datos a los consumidores.

¹⁹ http://www.geonovum.nl/sites/default/files/3d_pilot_artikel_engels_2.pdf

²⁰ <http://www.ingeoforum.de/en/portrait/about.html>

European COST Action TU0801²¹

Muchos de los modelos urbanos o ambientales se definen con el objetivo de ayudar a los profesionales e interesados en sus procesos de toma de decisiones. Los modelos que representan en 3D los elementos geométricos de una ciudad se utilizan cada vez más en diferentes ciudades y países para una amplia gama de aplicaciones más allá de la intención mera de visualización. Tales usos son posibles gracias a añadir información semántica a los aspectos geométricos. Por otra parte, en la perspectiva de un desarrollo sostenible, las ciudades deben ser estudiadas de manera integral teniendo en cuenta las muchas interrelaciones entre los diversos temas urbanos. Esto se puede lograr mediante la identificación y la extracción del conocimiento subyacente de datos y modelos relacionados. El uso de ontologías es una manera robusta para lograr el enriquecimiento semántico de los modelos 3D urbanos, así como su interoperabilidad con otros modelos urbanos, de modo que se conviertan en una matriz eficaz de los conocimientos urbanos en una perspectiva de sostenibilidad. En esta iniciativa se pretende (1) crear una plataforma de integración basada en los modelos 3D urbanos semánticamente enriquecidos, (2) utilizar una metodología basada en ontologías que podrían ser reutilizadas y (3) desarrollar y evaluar la usabilidad de una plataforma integrada para la planificación y toma de decisiones.

ESPRESSO

ESPRESSO²² tiene como objetivo el desarrollo de un *framework* de información de *smart city* que se construirá alrededor de CityGML como modelo de datos de referencia. Además, se codificarán servicios de datos para integrar y procesar los datos de manera eficiente en las aplicaciones de *smart city*. El proyecto también mapeará los estándares actuales, en línea con las posibles acciones de otros planes europeos y documentos de directrices. Se hará especial hincapié en el desarrollo de los componentes del *framework* que permitan la integración de datos ubicados geográficamente, ya que la información de la ubicación permite beneficios tecnológicos en las *smart city*. El enfoque de ESPRESSO

²¹ http://www.cost.eu/COST_Actions/tud/TU0801

²² <http://www.opengeospatial.org/projects/initiatives/espresso>

enfatisa la reducción de costes y fomentará un mercado abierto para muchos agentes, evitando el bloqueo de soluciones patentadas.

OGC Smart Cities Spatial Information Framework.

Los miembros del Open Geospatial Consortium (OGC) han aprobado el *white paper* del OGC Smart Cities Spatial Information Framework. Este documento está disponible de forma gratuita en https://portal.opengeospatial.org/files/?artifact_id=61188. Este documento aborda un marco de estándares de tecnologías de información abiertas que son fundamental a la hora de realizar las comunicaciones en las *Smart Cities*.

Este *white paper* de OGC proporciona las guías iniciales de un marco de información espacial que permita obtener inteligencia espacial urbana basándose en estándares abiertos tales como CityGML, IndoorGML, Moving Features y Augmented Reality Markup Language 2.0 (ARML 2.0). La idea de este marco de información espacial es proporcionar la base para integrar las características GIS, imágenes, observaciones de sensores y redes sociales que permitan gobernar y proporcionar servicios de la ciudad.

3.2.4 Generación y mantenimiento de modelos CityGML

El mayor problema de la generación y actualización de modelos CityGML es el elevado coste en tiempo y dinero asociado a la obtención de un modelo realista y de gran calidad [55]. Existen varias alternativas y enfoques para generar modelos CityGML, a continuación se describen algunas de ellas:

Uno de los métodos más rápidos es la generación procedural de ciudades [56]. Con este enfoque se pueden generar un gran número de ciudades a base de repetir estructuras y ajustar parámetros como la densidad de población o la edad de los edificios. Se trata de una opción útil para crear modelos CityGML para juegos y otras aplicaciones en las que el realismo no es prioritario.

Otra alternativa para crear modelos CityGML es el método basado en Open Street Map (OSM) [57]. Con él se puede generar el modelo geométrico tridimensional de una ciudad entera a partir de las contribuciones de los usuarios y del compromiso de los organismos locales. Sin embargo, requiere la participación de una comunidad de usuarios muy grande y no se puede garantizar la precisión de los datos aportados.

Otra fuente de datos igualmente útil para la generación de modelos CityGML son los datos del catastro [58]. En este caso la calidad de los datos es muy fiable, aunque no incluye información para la representación de detalles geométricos, alturas de los edificios, tipos de tejado, etc.

Los datos de LIDAR son otra fuente de información de gran valor para generar modelos CityGML [59]. Pese a que es posible crear un modelo realista de forma rápida a partir de la nube de puntos que estos datos facilitan, el resultado carece de la información necesaria para identificar los elementos urbanos (edificios, árboles, mobiliario, etc.), por lo que solo resulta útil para fines de mera visualización.

La estrategia más prometedora para la generación de un modelo CityGML es combinar los anteriores enfoques, usando datos del catastro u OSM para disponer la geometría de las plantas de los edificios y datos LIDAR para los detalles y alturas de los edificios [60][61]. También se puede integrar un modelo digital del terreno para poder ubicar el modelo en su elevación correcta.

Existen también una serie de herramientas que permiten crear modelos CityGML:

- **CityEditor**²³. CityEditor no es un programa independiente sino una extensión para Trimble SketchUp que permite importar modelos CityGML. Dispone de mecanismos de edición que permiten editar los atributos específicos de CityGML. También es posible exportar el resultado a un modelo CityGML, así como a un documento 3D-PDF, presentaciones web interactivas en 3D que pueden ser visualizadas en CityBrowser, y otros formatos.

A continuación, se presenta un ejemplo de la ciudad de Lund modelado en CityEditor y en el que cada superficie (paredes, tejados, suelos) están correctamente semantizados siguiendo el estándar CityGML (ver Figura 3-3).

²³ <https://www.3dis.de/loesungen/3d-stadtmodelle/cityeditor/>

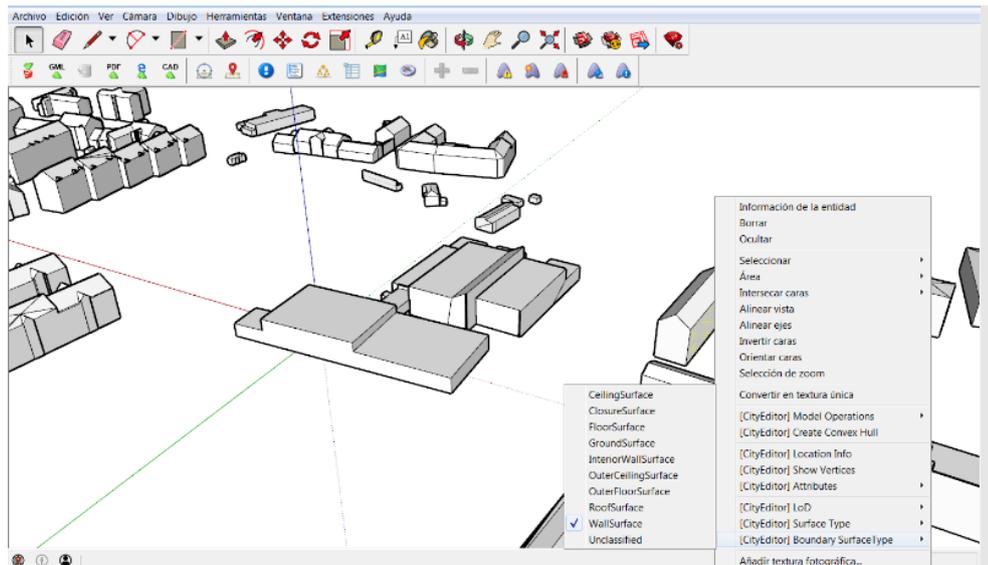


Figura 3-3 Ejemplo de la ciudad de Lund modelado en CityEditor

Pros: Al estar embebido dentro de SketchUp, el manejo y aprendizaje es fácil e intuitivo. Permite modelar edificios y también otros elementos de la ciudad.

Contras: Herramienta de pago. Es necesario disponer de SketchUp Pro para utilizar CityEditor. No está orientado a modelar zonas grandes, si no a zonas acotadas que pueden modelarse manualmente o la edición de modelos generados previamente con otras herramientas.

- **FME Desktop**²⁴. FME Desktop permite realizar conversiones, intercambio, transformación, validación e integración de datos. El FME desktop dispone de los siguientes componentes: 1) FME Workbench es un editor de flujo de trabajo visual que se utiliza para desarrollar herramientas de transformación de datos. 2) FME Data Inspector, que permite inspeccionar los atributos de los datos y la geometría. Esta herramienta hace que sea más fácil de entender la estructura de datos – antes, después y durante la traducción. 3) FME Quick Translator, que es una interfaz para realizar conversiones de formato.

A continuación, se presenta un ejemplo de cómo generar un modelo CityGML en FME a partir de un fichero ESRI Shapefile (SHP) (ver Figura 3-4):

²⁴ <https://www.safe.com/fme/fme-desktop/>

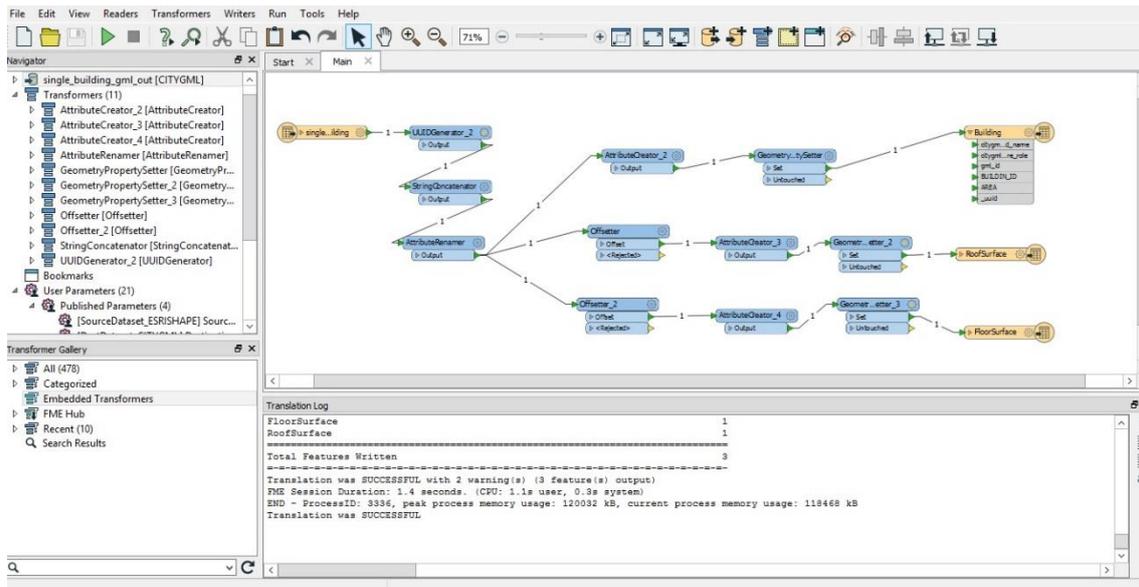


Figura 3-4 Ejemplo de cómo generar un modelo CityGML en FME a partir de un fichero SHP

Pros: Está orientado a modelar zonas grandes. Una vez se conoce la herramienta, es fácil de crear y modificar workflows. Permite trabajar con múltiples ficheros (y de gran tamaño) a la vez. Permite modelar edificios y también otros elementos de la ciudad.

Contras: Herramienta de pago. Se necesita tiempo para aprender a utilizar la herramienta.

- **BuildingReconstruction**²⁵. BuildingReconstruction es un software para la extracción automática de modelos de edificios en 3D a partir de modelos de superficie digitales. BuildingReconstruction es adecuado para todos los usuarios que desean crear de manera eficiente modelos de construcción 3D a partir de modelos digitales de elevación. La alta tasa de reconocimiento de la reconstrucción automática permite la captura correcta de la geometría del edificio en la mayoría de los casos. El procesamiento posterior de las estructuras restantes del edificio y del techo es ayudado por un editor fácil de usar que permite la edición manual, semiautomática o completa.

Pros: Está orientado a modelar zonas grandes. Permite trabajar con múltiples ficheros (y de gran tamaño) a la vez. Permite editar los modelos de forma manual.

²⁵ <http://www.virtualcitysystems.de/en/products/buildingreconstruction>

Identifica la curvatura de los tejados. Permite modelar edificios y también otros elementos de la ciudad.

Contras: Herramienta de pago.

- **3dfier**²⁶. Toma conjuntos de datos 2D GIS (por ejemplo, conjuntos de datos topográficos) y los hace tridimensionales al elevar cada polígono a 3D. La elevación se obtiene de una nube de puntos LIDAR (utilizando ficheros LAS / LAZ), y la semántica de cada polígono se usa para realizar el levantamiento. Es decir, los polígonos de agua se extruyen a los polígonos horizontales, los edificios a los bloques LOD1, las carreteras como superficies lisas, etc. Cada polígono se triangula mediante la triangulación Delaunay limitada.

Pros: Herramienta gratuita. Está orientado a modelar zonas grandes. Permite modelar edificios y también otros elementos de la ciudad.

Contras: No identifica la curvatura del tejado. Crea demasiados polígonos, siendo más complicado el posterior uso.

A continuación, se presenta el resultado en CityGML utilizando la herramienta 3dfier (ver Figura 3-5):

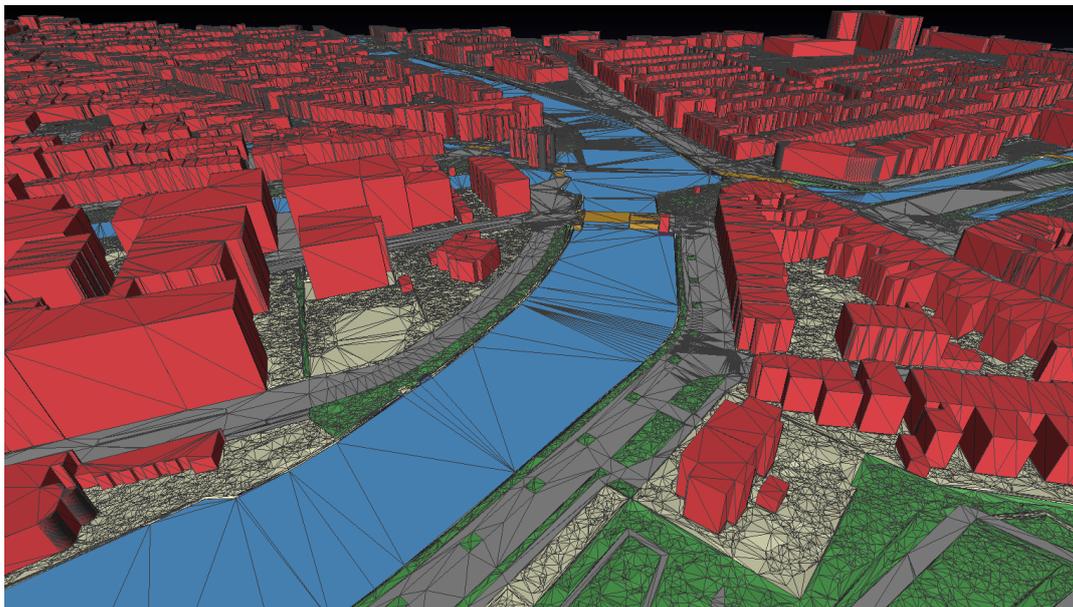


Figura 3-5 Resultado en CityGML utilizando la herramienta 3dfier (<https://github.com/tudelft3d/3dfier>)

²⁶ <https://github.com/tudelft3d/3dfier>

- **MOSS. novaFactory²⁷.** novaFACTORY es una solución para datos de geotopografía y servicios basados en ArcGIS Server. Ya sean ortofotos, mapas topográficos, modelos de terreno, modelos de paisaje o edificios en 3D, novaFACTORY permite aumentar la eficiencia de los procesos de trabajo.
Pros: Está orientado a modelar zonas grandes. Identifica la curvatura de los tejados.
Contras: Herramienta de pago. Para que la herramienta funcione correctamente hay que adecuar los datos totalmente a sus especificaciones. Genera únicamente edificios.

Ninguna de las herramientas descritas anteriormente cumple todos los requisitos definidos en el capítulo anterior. Hay herramientas que se centran únicamente en los edificios, sin tener en cuenta el resto de elementos urbanos. Otros que, aunque si tienen en cuenta otros elementos urbanos además de los edificios, no están pensadas para realizar la generación de forma masiva ya que hay que realizar el modelado de forma manual. Estas herramientas no permiten la semantización de forma natural, ni tampoco la realización de inferencias y la obtención de datos semánticos procesando la geometría. Además, tampoco permiten realizar un mantenimiento continuo de una forma natural.

Después de generar y volcar la información semántica al modelo de datos, se requiere un mantenimiento continuo para tenerlo al día. Este mantenimiento afecta tanto a los datos geométricos como a los semánticos [62][63][64]. Como explicaban en [26], las tareas de mantenimiento a menudo consisten en recabar datos históricos de diferentes fuentes y actualizar la base de datos con ellos.

Otra posibilidad es mantener el modelo de datos de forma automática, por medio de la reconstrucción por componentes [27]. En este trabajo combinaron información proveniente de diferentes sensores y mapas de un sistema de información geográfica con un enfoque semántico y basado en componentes para el modelado y reconstrucción de edificios complejos.

Otro de los componentes importantes a la hora de realizar el mantenimiento de un modelo de datos urbano es representar la evolución en el tiempo o presentar diferentes escenarios

²⁷ <https://www.moss.de/novafactory/>

o alternativas de diseño para un proyecto de transformación urbanística. El esquema CityGML actual no es compatible con la definición de variantes o escenarios cambiantes.

Una de las principales limitaciones de CityGML es que los valores de los atributos se establecen solo una vez. Eso significa que, cuando se actualiza el valor de un atributo, se pierde el valor anterior. Este problema, además del versionado, se identificó desde el comienzo de CityGML [65]. Como no hay gestión de datos históricos, solo se representan los valores del estado actuales [66]. Para ello, proponen una extensión que permite considerar cambios de forma frecuente, como es el caso de los datos de los sensores que pueden vincularse con objetos de la ciudad. Así pues, los requisitos de los modelos 3D urbanos han aumentado, lo que significa que además de la información geométrica, existe también necesidad de disponer de información semántica y datos actualizados [66]. Otra limitación es que el usuario no sabe cuándo se ha establecido un atributo, con lo cual, el valor puede ser el valor actual o uno antiguo. Para ello se necesita una dimensión adicional que permita representar los cambios a lo largo del tiempo [67].

Existen alternativas tales como los dinamizadores, que tienen como objetivo implementar propiedades dinámicas para modelos 3D urbanos [68]. Varias áreas de aplicación de modelos 3D urbanos, como las simulaciones, requieren el uso de atributos altamente dinámicos y variables en el tiempo. Sin embargo, incluso si con los dinamizadores es posible representar los valores semánticos a lo largo del tiempo, todavía no es posible representar escenarios cambiantes o variantes. Se han definido diferentes acercamientos hacia las variantes desde la adopción del estándar CityGML. Una de las alternativas es definir una variante adicional para cada LOD [69]. En esta definición de variantes se centran en aspectos geométricos del edificio. Sin embargo, no se centran en los datos semánticos y en escenarios cambiantes. Como ejemplo, en [70] han definido variantes en la fase de diseño de edificios. Para ellos, las variantes también pueden describir diferentes estados de planificación de situaciones temporales del edificio. Aunque de esta forma es posible cambiar algunas de las características de los objetos de la ciudad; sigue sin poder representar el cambio de tipos de objetos de la ciudad. Otra alternativa para representar escenarios cambiantes es creando un modelo específico de CityGML para cada variante [71]. Sin embargo, no es una solución adecuada si la cantidad de las variantes necesarias es grande.

En la sección 3.3.2 se plantea una aproximación para representar variantes dentro de un modelo CityGML que tiene en cuenta los siguientes aspectos principales:

- La cantidad de escenarios o variantes que se deben poder representar puede ser grande.
- Copiar todo el fichero CityGML para cada variante no es una solución factible.
- Debe ser posible representar cambios de tipos de elementos de la ciudad. P.ej. cambiar un área verde por un edificio.
- Debe ser posible representar cambios en las características de los diferentes elementos de la ciudad. P.ej. Cambiar la función de un edificio de edificio industrial a residencial, o cambiar el número de plazas de aparcamiento que tiene un parking.
- Debe ser posible representar la creación o eliminación de elementos urbanos representados mediante *Point of Interest* (POI), como paradas de autobús, paradas de tranvía, estaciones de tren, semáforos o áreas de juegos, entre otros.

3.3 Ciclo de vida de ciudades en 3D

El objetivo de esta sección es describir un flujo de trabajo para la gestión de un modelo 3D urbano a lo largo de todo su ciclo de vida, incluida la creación de modelos urbanos tridimensionales basados en CityGML a partir de diferentes fuentes de datos. Dicho flujo de trabajo sirve para generar edificios y otros elementos urbanos (tales como vegetación, mobiliario urbano o carreteras, entre otros) y complementa el modelo de datos con información semántica.

El ciclo de vida de los modelos 3D urbanos es similar al de otros contenidos digitales, en los cuales es necesario: conceptualizar, crear y recibir datos, evaluar y seleccionar, preservar, almacenar, acceder, utilizar y reutilizar y, finalmente, transformar [72].

El ciclo de vida de las ciudades en 3D se puede dividir en tres tareas principales: generación, uso y mantenimiento (ver Figura 3-6).

- **Generación:** Abarca la identificación y obtención de datos desde diferentes fuentes, así como la configuración de los modelos de datos con parámetros temáticos específicos y la generación de un modelo tridimensional básico (incluida la geometría y la semántica) con edificios en LOD0, LOD1 y LOD2 y

otros datos adicionales, tales como el modelo digital del terreno con ortofotos, carreteras, vegetación o puntos de interés (POI, *Points of Interest*), entre otros.

- **Uso:** Permite aprovechar el modelo urbano que se ha generado con ayuda de aplicaciones para la visualización 3D interactiva. El uso de los modelos se ha abordado en el capítulo anterior (sección 2.3).
- **Mantenimiento:** Para poder disponer de modelos 3D urbanos actualizados y reales, es importante llevar a cabo diferentes procesos de mantenimiento de manera periódica. Estos procesos permiten actualizar y añadir información geográfica y semántica al modelo 3D urbano.

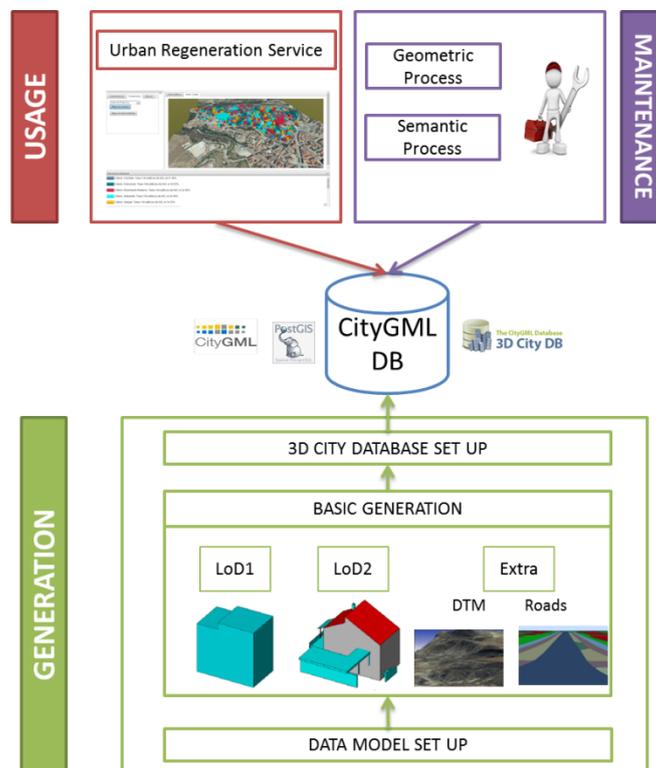


Figura 3-6 Ciclo de vida del modelo 3D urbano

3.3.1 Workflow de generación

Se ha definido un flujo de trabajo detallado destinado a la fase de generación del modelo 3D urbano. La Figura 3-7 muestra un diagrama de actividades UML de dicho flujo de trabajo. Los diagramas de actividades expresan procesos en forma de secuencias de acciones. Cada actividad engloba diferentes acciones y decisiones relativas a las tareas y tienen un punto de inicio y otro de final. Una acción (que se ilustra a modo de cápsula) representa un paso individual, es decir, no se puede desglosar más, dentro de una actividad. Los nodos de decisión se muestran en forma de diamantes y los almacenes de datos, que son nodos de memoria intermedia de información no transitoria, son cajas estereotipadas. Para más información sobre diagramas de actividades UML, véase [73].

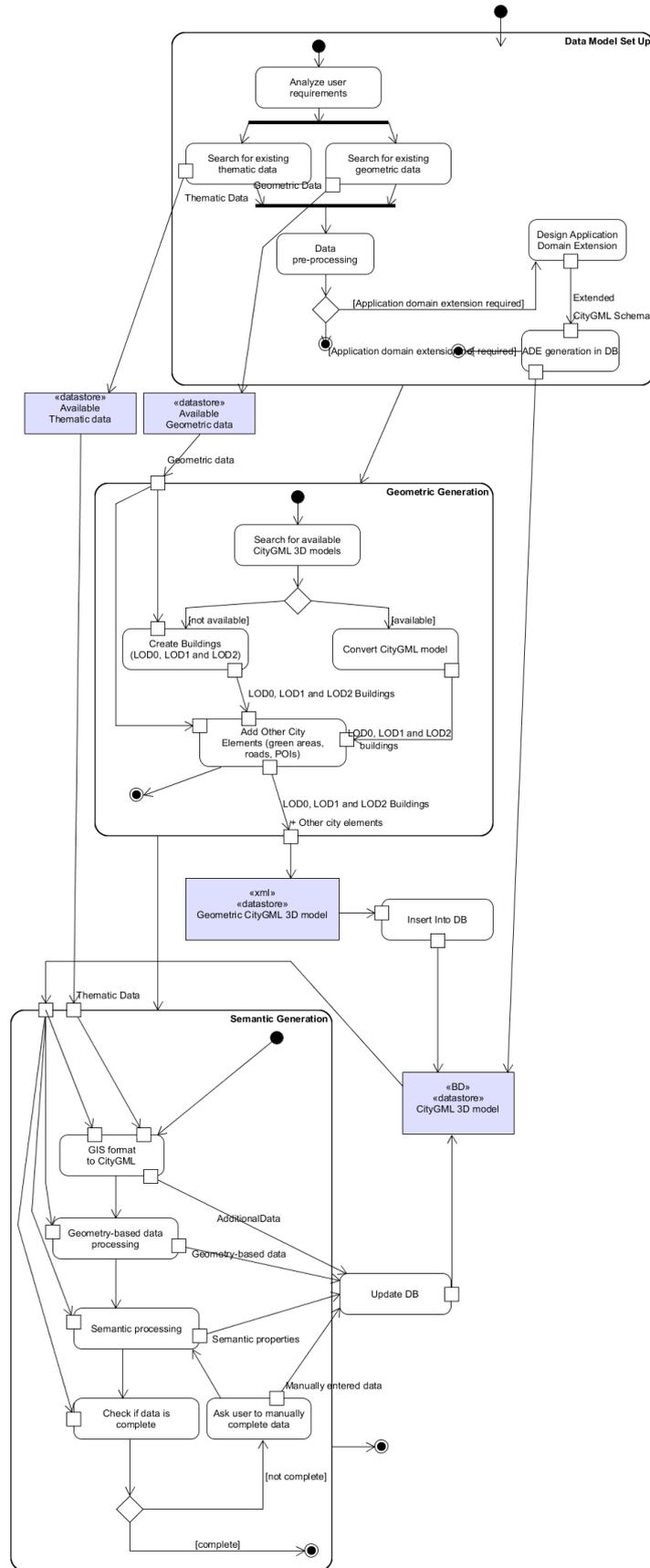


Figura 3-7 Flujo de trabajo de generación del modelo 3D urbano

El flujo de trabajo se divide en tres actividades principales: 1) la configuración del modelo de datos, que es donde se estudian los requisitos del usuario, se buscan datos geométricos y semánticos ya disponibles, se preparan dichos datos para adaptarlos a las necesidades del proceso y se diseñan las extensiones de dominio de aplicación (ADE); 2) la generación de la geometría, es donde se generan las geometrías de los edificios en diferentes LOD, así como las zonas verdes, las carreteras y demás elementos urbanos; y, finalmente, 3) la generación de la semántica, que es donde se completa la información semántica a partir de fuentes de datos externas, ya sea de forma automática o manualmente. La información tanto geométrica como semántica queda almacenada en CityGML, lista para ser actualizada cuando sea necesario. El modelo puede almacenarse en paralelo en una base de datos geoespacial para facilitar las consultas y actualización de dicho modelo.

3.3.1.1 Configuración del modelo de datos

(1) *Analyze user requirements*

La primera acción de la Configuración del modelo de datos ("*Data Model Set Up*") es "Analizar los requisitos del usuario" ("*Analyze user requirements*"), lo que implica identificar toda la información temática necesaria para el caso de estudio, teniendo en cuenta los diferentes objetos de la ciudad.

Existen herramientas tales como ReqCap [74] que permiten identificar los requisitos de usuario de una manera ordenada y evitando requisitos duplicados o mal descritos. De esta forma, aun habiendo diferentes agentes involucrados a la hora de identificar los requisitos de usuario, se facilita la recogida de los requisitos.

Como resultado de esta acción se obtiene un listado de parámetros de los cuales se almacena el nombre, descripción, tipo, escala y valores posibles, entre otros. Así se obtiene un diccionario de datos que recopila los requisitos de usuario para el modelo CityGML.

(2) *Search for existing thematic data / Search for existing geometric data*

A continuación, se pueden ejecutar dos acciones en paralelo: la "Búsqueda de información temática disponible" ("*Search for existing thematic data*") y la "Búsqueda de datos geométricos disponibles" ("*Search for existing geometric data*"). En ambos casos,

el objetivo es identificar las fuentes de datos existentes útiles para la obtención de información geométrica y semántica para el área de estudio en cuestión.

Las principales fuentes geométricas son, siempre con relación al área seleccionada, las plantas de los edificios, información de los edificios en formato SHP (además de la geometría en sí, datos como el número de pisos también pueden ser aprovechables para la generación de la geometría), modelos digitales del terreno (DTM), ortofotografías (para texturizar los DTM y los tejados) y datos LIDAR que faciliten la altura de los edificios, texturas de las fachadas y modelos 3D detallados.

Algunos de estos conjuntos de datos están accesibles al público gratuitamente o sujetos a tasas, dependiendo del país o incluso de la región. Las principales fuentes de datos son: los catastros nacionales y regionales, los institutos geográficos nacionales y regionales, Google, Microsoft y OpenStreetMap.

(3) Data pre-processing

Por lo general, estos conjuntos de datos deben ser preprocesados como paso previo a la generación del modelo CityGML. A la hora de adaptar la información recogida de fuentes de datos originales al formato necesario para la generación geométrica y semántica hay tres tareas esenciales: 1) el filtrado, 2) la simplificación y 3) la limpieza. Para ello es necesario, por ejemplo, filtrar los elementos geométricos que no representan edificios (piscinas, patios, jardines, terrazas, etc.), eliminar elementos antiguos o duplicados y agrupar las geometrías que representan un único edificio.

Las líneas que representan otros elementos urbanos tales como zonas verdes o ríos deben transformarse en polígonos cerrados. Además, todas las capas de información deben representarse dentro del mismo sistema de referencia. Otro ejemplo de preprocesado consiste en asociar la información semántica disponible a nivel de portal (normalmente representado por un punto) a un edificio. La dificultad viene dada ya que suelen existir varios portales asociados a una única geometría, con lo cual el proceso de unir la información suele ser complejo.

A continuación, se describen una serie de posibles preprocesamientos que resultan habituales:

- La información que proviene de diferentes fuentes de datos no dispone de un identificador compartido, lo que significa que deben realizarse diferentes geo-procesos para vincular o asociar datos de diferentes fuentes de datos.
- La información no está incluida en una única capa, por lo que es necesario mezclar varias capas (como parcelas y edificios).
- Inconsistencias entre los datos. Tales como edificios con 20 m de altura y solo 1 planta.
- Las filas sin datos tienen que descartarse para evitar errores posteriores.
- La información semántica no está completa para todos los edificios.
- Las capas generadas a partir de datos CAD generalmente incluyen únicamente información geométrica, no incluyen información semántica.
- Los sistemas de referencia de las capas son diferentes.
- Hay capas en las que cada polígono no representa un edificio, ya que representan diferentes partes del edificio que tienen diferente altura. Por lo tanto, estos polígonos deben eliminarse o unir todos en un único polígono.
- Algunas veces la geometría de los edificios es demasiado compleja para los propósitos del proyecto (p.e. un polígono que representa una planta de un edificio con más de 100 vértices).
- No hay un acuerdo común sobre la representación de algunas de las propiedades del edificio, como el uso del edificio, con lo cual es necesario adaptarlo.

(4) Application domain extension required

Una vez identificadas todas las fuentes de datos, es probable que sea necesario crear una ("Extensión de dominio de aplicación requerida" ["*Application domain extension required*"]). En tal caso, se procede a diseñar una ADE ("Diseñar extensión de dominio de aplicación" ["*Design Application Domain Extension*"]), incluyendo todos los datos exigidos por el usuario que no figuren en el *core* de CityGML. El resultado será un "Esquema CityGML extendido" ("*Extended CityGML Schema*") en formato XSD (*XML Schema Definition*). En la última acción, "Generación de ADE en la base de datos" ("*ADE generation in DB*"), la estructura del modelo de datos se modifica/adapta para integrar los datos de la ADE.

El estándar CityGML describe en la especificación cómo representar los datos adicionales no contemplados en el *core* como una extensión de dominio de aplicación (ADE). En la especificación se describe cómo realizar una ADE para adaptar el esquema XSD de CityGML. Sin embargo, no se explica cómo diseñar una ADE en UML. Con lo cual, el grupo de Van den Brink crearon la extensión IMGeo ADE modelado en UML. La ventaja es que los esquemas XSD pueden generarse automáticamente a partir del UML [75]. Para ello, documentaron sus experiencias y lecciones aprendidas en un documento con la intención de hacer de esto una mejor práctica.

3.3.1.2 Generación de la geometría

La siguiente actividad es la Generación de la geometría ("Geometric Generation"). En ella, los únicos datos de entrada son los "Datos geométricos disponibles" ("Available Geometric data") identificados en la actividad anterior.

(5) Search for available CityGML 3D models

A continuación, se lleva a cabo una "Búsqueda de modelos 3D CityGML disponibles" ("Search for available CityGML 3D models"). Si ya existe un modelo CityGML, este debe someterse a la tarea de "Convertir el modelo CityGML" ("Convert CityGML model"), para adaptarlo a los requisitos de generación específicos (es decir, versión CityGML, estructura jerárquica de los elementos del edificio, etc.).

Actualmente hay disponibles modelos CityGML de muchas ciudades. Normalmente están accesibles haciendo un ZIP de los modelos o subiéndolos directamente a un servidor de ficheros. Algunos ejemplos son: algunas ciudades holandesas²⁸, Bruselas²⁹, Hamburgo³⁰, Linz³¹, Montreal³² o New York^{33 34}, entre otros. Hay una gran variación de las características en los anteriores modelos CityGML. Algunos contienen pocos edificios, mientras que otros representan más de cien mil edificios. Hay modelos CityGML que se representan en un único fichero, mientras otros están divididos en cuadrantes. También

²⁸ <https://3d.bk.tudelft.nl/opendata/3dfier/>

²⁹ <http://urbisdownload.gis.irisnet.be/en/dimension>

³⁰ <http://suche.transparenz.hamburg.de/dataset/3d-stadtmodell-LOD2-de-hamburg1>

³¹ http://geo.data.linz.gv.at/katalog/geodata/3d_geo_daten_LOD2/

³² <http://donnees.ville.montreal.qc.ca/dataset/maquette-numerique-batiments-citygml-LOD2-avec-textures>

³³ <http://www.gis.bgu.tum.de/en/projects/new-york-city-3d/>

³⁴ <http://www1.nyc.gov/site/doitt/initiatives/3d-building.page>

difieren a la hora de representar otros elementos de la ciudad, aunque la mayoría se centran únicamente en los edificios.

(6) *Create Buildings (LOD0, LOD1 and LOD2)*

Si, por el contrario, aún no existe ningún modelo 3D urbano en CityGML, se ejecuta en su lugar la tarea "Crear edificios (LOD0, LOD1 y LOD2)" ("*Create Buildings (LOD0, LOD1 and LOD2)*"). Este paso del proceso se basa en un desarrollo propio de generación de modelos CityGML a partir de la combinación de la geometría de las plantas de los edificios (proporcionada en formato SHP) y los datos LIDAR del entorno a modelar. El proceso es el siguiente:

- **Procesar los ficheros LIDAR.** Es necesario procesar todos los ficheros LIDAR, tanto del DTM como del *Digital Surface Model (DSM)*. En este caso, se utiliza el formato ASCII Raster format (.ASC). Primero se identifica cual es el *bounding box* que representa cada fichero LIDAR. De esta forma, después, a la hora de necesitar identificar la altura de un punto en concreto, es posible saber a qué fichero LIDAR hay que acceder.
- **Obtener altura del DTM y DSM para cada edificio.** Se procesan todos los puntos de los ficheros LIDAR (DTM y DSM) y se identifica si el punto en cuestión interseca con la geometría de alguno de los edificios. En caso afirmativo, el valor de la altura de ese punto se asocia con el identificador de ese edificio. Este proceso se repite con todos los puntos de todos los ficheros LIDAR.
- **Obtener altura para cada edificio.** Al finalizar el proceso anterior disponemos por cada edificio, los puntos del DTM y DSM que intersecan dentro del mismo. Después, se eliminan los datos atípicos, eliminando los puntos por debajo del cuartil 5 y los que están por encima del cuartil 95. A continuación, se obtiene la media aritmética del resto de valores. Repitiendo este proceso para cada edificio, obtenemos la altura media respecto al DTM y DSM para cada edificio. Y la altura del edificio se obtiene restando el valor medio del DSM con el valor medio del DTM. Como este proceso conlleva almacenar una gran cantidad de datos durante cierto tiempo, el número de edificios se divide por lotes, agilizando así la generación.
- **Generar el LOD0 de cada edificio en CityGML.** Para eso, por cada polígono que se dispone para cada edificio, se procesa cada punto del polígono y se obtiene

la altitud de ese punto en los ficheros LIDAR del DTM. Disponiendo de la altitud de cada punto del polígono es posible generar el LOD0 en CityGML.

- **Generar el LOD1 de cada edificio en CityGML.** Para la generación del LOD1 el proceso es diferente. Se procesan los puntos del polígono del edificio de dos en dos. Es decir, el primer punto y el segundo punto crean la primera fachada; el segundo punto y el tercero crean la segunda fachada; y así sucesivamente. De esta forma se obtienen fachadas o líneas de dos puntos para cada edificio. Al igual que con el LOD0, se obtiene la altitud de los puntos en el DTM y para generar la fachada se realiza la extrusión utilizando la altura media obtenida anteriormente. Los tejados se representan de forma plana. De esta forma es posible generar el LOD1 en CityGML.
- **Generar el LOD2 de cada edificio en CityGML.** Para la generación del LOD2 el proceso es casi igual que el LOD1, con la diferencia que en el LOD2 es necesario indicar que tipo de superficie se genera en CityGML. Si es un suelo (GroundSurface), fachada (WallSurface) o tejado (RoofSurface). La identificación del tipo de superficie representada se realiza en función del ángulo de la normal de dicha fachada (polígonos verticales representan fachadas y horizontales representan suelo cuando la altura es igual o próxima a la altitud media del DTM y tejado cuando la altura es igual o próxima a la altitud del DSM).
- **Generar el elemento building en CityGML.** Después, se genera por cada edificio un elemento building en CityGML y se añaden las geometrías en LOD0, LOD1 y LOD2.

En la Figura 3-8 se resume de forma gráfica el proceso seguido, las fuentes de datos necesarias y el resultado obtenido. También se incluye la generación de otros elementos urbanos que se presenta en el siguiente paso:

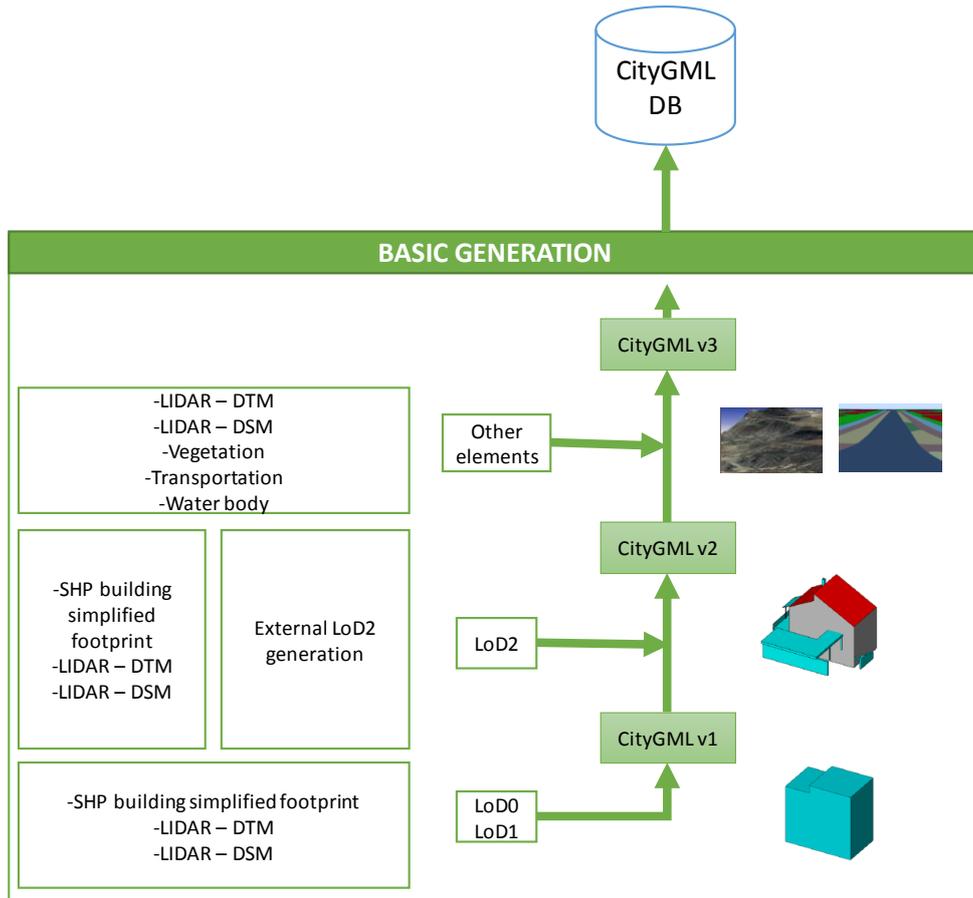


Figura 3-8 Create Buildings (LoD0, LOD1 and LOD2)

(7) Add Other City Elements (green areas, roads, POIs, 3D models)

Con la siguiente acción, "Añadir otros elementos urbanos (zonas verdes, carreteras, POI, modelos 3D)" ("Add Other City Elements (green areas, roads, POI, 3D models)") se generan elementos urbanos adicionales.

- El DTM aparece representado como mallas TIN (red irregular triangulada o *Triangulated Irregular Network*).
- Las carreteras se generan en LOD0 (solo líneas) y se asignan a la capa de transporte de CityGML.
- Las zonas verdes, por su parte, son creadas y asignadas al modelo temático de vegetación de CityGML.
- Las zonas de agua son creadas y asignadas a los *water body* de CityGML.
- Del mismo modo, en este punto cualquier modelo 3D de un edificio obtenido a partir de una fuente externa, como CAD, SketchUp o IFC, se convierte en un modelo CityGML.

- También es posible añadir al modelo CityGML los **POI** a modo puntos.

Como resultado se obtiene un archivo CityGML con diferentes LOD: "Edificios LOD0, LOD1 y LOD2 + Otros elementos urbanos" ("LOD0, *LOD1 and LOD2 Buildings + Other City Elements*"). LOD0 incluye DTM, carreteras y zonas verdes, mientras que las geometrías de los edificios están en LOD0, LOD1 y LOD2.

(8) *Insert into DB*

CityGML está concebido para representar la información semántica y geométrica de objetos urbanos. Si bien XML es el formato recomendado para intercambiar esta información, no es la mejor opción para guardar y recuperar modelos 3D urbanos complejos que contengan miles de objetos, ya que estos archivos pueden pesar varios gigabytes.

Una alternativa es guardar la información en una base de datos relacional ampliada con capacidades espaciales. PostgreSQL + PostGIS es una opción válida para ello, ya que es compatible con objetos tridimensionales georreferenciados. Existen otras alternativas tales como Oracle Spatial que también permite representar objetos tridimensionales georreferenciados en una base de datos.

En este *workflow* se utiliza la base de datos 3DCityDB³⁵ que permite almacenar modelos CityGML en una base de datos PostgreSQL + PostGIS. 3DCityDB ofrece tanto los scripts de generación de la base de datos como las librerías que permiten tanto importar a la base de datos como exportar desde la base de datos modelos CityGML.

Los datos resultantes de la generación geométrica del modelo CityGML se guardan en la base de datos en la actividad "Insertar en la base de datos" ("*Insert into DB*"). Además, se despliegan también los servicios WFS para poder acceder a los datos de la base de datos realizando peticiones de forma estándar.

3.3.1.3 *Generación semántica*

La última actividad es la "Generación semántica" ("*Semantic Generation*"), en la que el modelo de datos se completa con datos temáticos. Esta actividad, que toma como datos de entrada los "Datos temáticos" ("*Thematic Data*") obtenidos durante la primera

³⁵ <https://www.3dcitydb.org>

actividad y la "Base de datos de modelos 3D CityGML" ("*CityGML 3D model DB*"), consta de 4 acciones principales.

El resultado, una vez completadas todas las acciones, se guarda directamente en la base de datos de modelos 3D CityGML, lo que corresponde a la acción "Actualizar base de datos" ("*Update DB*").

(9) *GIS format to CityGML*

El objetivo de la acción "De formato GIS a CityGML" ("*GIS format to CityGML*") es añadir información semántica al modelo. Para ello, se ejecuta un proceso ETL de exportación, transformación y carga (*Extract Transform Load*) con una coincidencia entre el formato GIS y los parámetros CityGML. De esta manera es posible dotar de contenido semántico de forma masiva a los elementos geométricos generados en la actividad anterior.

(10) *Geometry-based data processing*

La acción "Procesamiento de datos basados en la geometría" ("*Geometry-based data processing*") emplea la "Base de datos de modelos 3D CityGML" ("*CityGML 3D Model DB*") como datos de entrada.

El objetivo es procesar la geometría del modelo para poder obtener nueva información. Por ejemplo, es posible estimar el área y volumen del edificio. del mismo modo es posible analizar que fachadas son medianeras y cuáles no, sabiendo, de esta forma, que fachadas son exteriores. Este dato es muy interesante a la hora de realizar cálculos de eficiencia energética. De la misma forma, es posible obtener la orientación principal de las fachadas, obteniendo así el área de fachada por orientaciones. Al final de esta acción se obtienen valores tales como el área, el volumen, la orientación principal o la identificación de fachadas adyacentes.

(11) *Semantic processing*

La siguiente acción es la de "Procesamiento semántico" ("*Semantic processing*"), que permite deducir u obtener información semántica a partir de otros parámetros disponibles en el modelo de datos. Otros datos de entrada útiles para esta acción son archivos que incluyan reglas y condiciones.

Por ejemplo, el material principal del edificio se podría deducir a partir de la información de su año de construcción. Los edificios construidos antes de 1900 podrían asociarse con la piedra como principal material de construcción, los correspondientes al periodo 1900-1930, con el hormigón, y así con otros periodos. Otro ejemplo de completado semántico puede consistir en sumar el área de todos los edificios de un distrito, para saber cuál es el área edificada en ese distrito.

(12) *Check if data is complete*

La última acción (tanto para esta actividad concreta como para todo el flujo de trabajo en su conjunto) es asegurarse de que no se necesita ninguna otra información con la acción "Comprobar que los datos están completos" ("*Check if data is complete*"). Si no es así, el usuario debe completar los datos que faltan por medio de un formulario en la aplicación con la acción "Solicitar al usuario que complete los datos manualmente" ("*Ask user to manually complete data*").

Las implementaciones de los servicios de las acciones anteriormente explicadas se describen en la sección 3.4.

3.3.2 Operaciones de mantenimiento

Es muy habitual que los modelos 3D urbanos creados acaben quedando desactualizados. Sin embargo, a fin de poder hacer un uso realista de los modelos de datos, es imprescindible realizar algunas tareas para mantenerlos "vivos". Por eso, los modelos de datos deberían someterse a actualizaciones frecuentes de la información semántica y geométrica.

Mientras que las geometrías de edificios ya existentes deberían actualizarse por otras más detalladas o texturizadas de manera habitual, la frecuencia del resto de actualizaciones, como modificaciones geométricas más complejas o menos usuales (p. ej. la adición de edificios no existentes), debería ser menor.

Algunas de las acciones del flujo de trabajo descrito en la sección anterior pueden repetirse periódicamente para mantener los datos actualizados. Prácticamente todas las acciones representadas en el flujo de trabajo pueden llevarse a cabo durante el mantenimiento de los modelos, sin embargo las acciones más habituales durante el mantenimiento son: "Crear edificios, añadir otros elementos urbanos" ("*Create Buildings,*

Add Other City Elements"), "De formato GIS a CityGML" ("*GIS format to CityGML*"), "Procesamiento de datos basados en la geometría" ("*Geometry-based data processing*"), "Procesamiento semántico" ("*Semantic processing*") y "Solicitar al usuario que complete los datos manualmente" ("*Ask user to manually complete the data*").

Además de las acciones ya presentadas en el flujo de trabajo de generación que pueden repetirse durante el mantenimiento, se han identificado otras funcionalidades específicas de dicha etapa del ciclo de vida de un modelo CityGML. Algunas de dichas funcionalidades se describen a continuación.

Concatenar varios modelos CityGML

Con esta funcionalidad es posible unir toda la información contenida en dos ficheros CityGML en un único fichero. De esta forma es posible que dos personas trabajen en diferentes barrios de una ciudad y después juntar la información. O en caso de querer incluir un nuevo edificio de reciente construcción poder hacerlo también.

En la Figura 3-9 se presenta el ejemplo de unión de dos ficheros, el primero con varios edificios de un barrio y el segundo con un edificio que no estaba en el primero, pero que era necesario incluirlo.

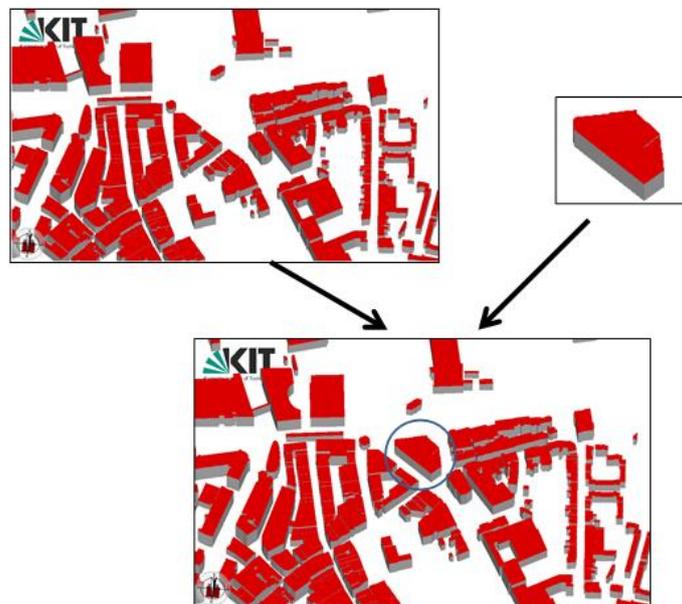


Figura 3-9 Operación de concatenar varios modelos CityGML

Reemplazar un edificio por otro en un modelo CityGML

Esta funcionalidad permite, a partir de dos ficheros de CityGML de entrada, reemplazar toda la información de un edificio concreto. Siendo el primer fichero CityGML el que contiene toda la información de un barrio/ciudad y el segundo fichero CityGML el que contiene la información de un único edificio (el que se quiere reemplazar), esta funcionalidad busca el edificio en cuestión y reemplaza la información ya disponible por la información del segundo fichero CityGML.

Mediante esta funcionalidad es posible adaptar la geometría de un edificio o incluso añadir la geometría en LOD3 de forma sencilla (ver Figura 3-10).

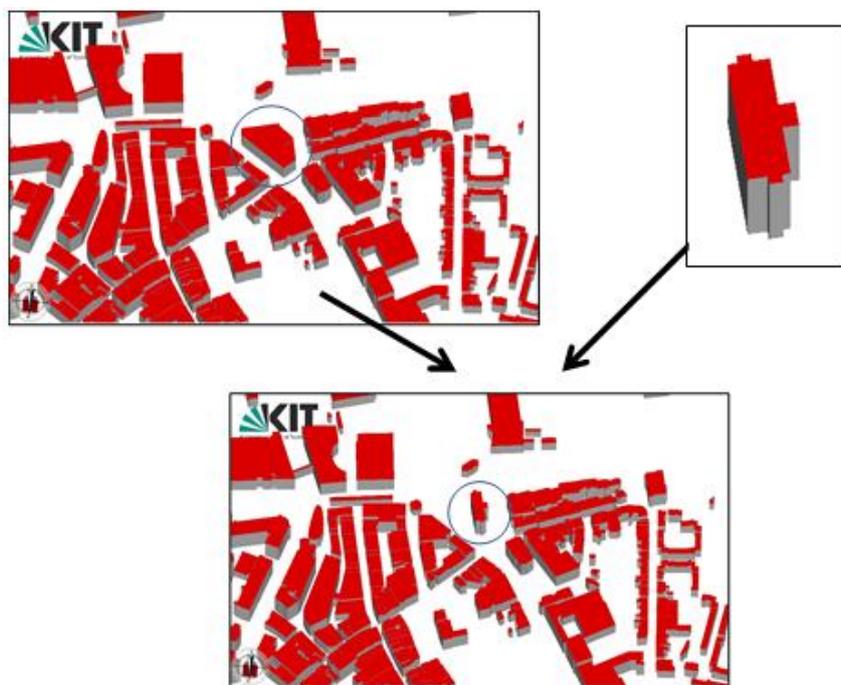


Figura 3-10 Operación de reemplazar un edificio por otro en un modelo CityGML

Seleccionar un conjunto de elementos de una zona del modelo CityGML

Esta funcionalidad permite filtrar los edificios que están a una distancia concreta respecto a un edificio. Indicando un fichero CityGML, el identificador de un edificio y una distancia en metros, es posible obtener los edificios alrededor del edificio seleccionado en la distancia concreta deseada.

Esta funcionalidad es útil por ejemplo a la hora de realizar simulaciones de eficiencia energética de un edificio en concreto, donde es necesario disponer de la geometría de los

edificios cercanos de cara a obtener el sombreado que provocan los edificios cercanos sobre un edificio en cuestión (ver Figura 3-11).

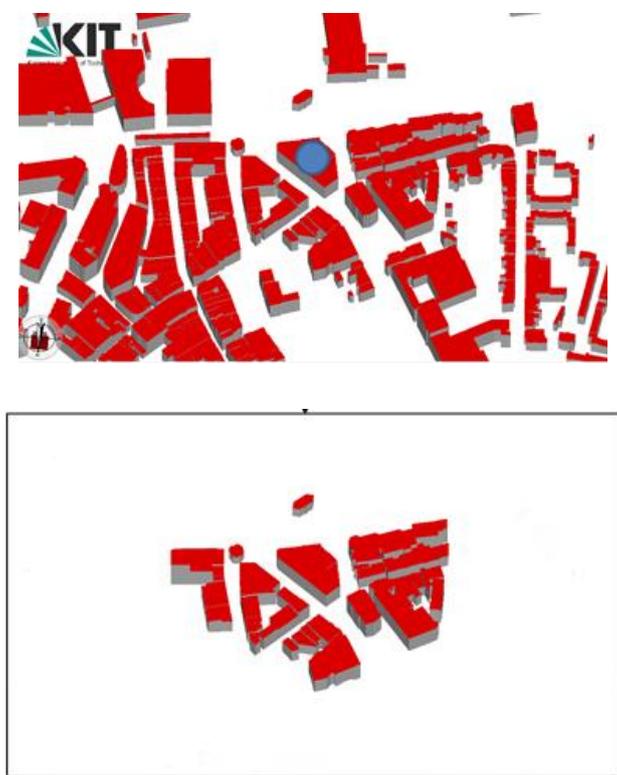


Figura 3-11 Operación de seleccionar un conjunto de elementos de una zona del modelo CityGML

Completar el modelo CityGML con un modelo de un edificio más detallado

Esta funcionalidad permite exportar un edificio en formato IFC a CityGML tanto a LOD2 como a LoD3. Esta funcionalidad es muy útil para completar el nivel de detalle de los edificios del modelo CityGML generados de forma masiva mediante el flujo de trabajo de generación.

El concepto BIM cada vez está más extendido con lo cual cada vez existen más edificios modelados en IFC. Poder incluir la información de estos modelos con detalle en formato IFC dentro del modelo CityGML es de gran utilidad (ver Figura 3-12).



Figura 3-12 Completar el modelo CityGML con un modelo de un edificio más detallado

Cambiar el tipo de elemento urbano

En este caso, es posible cambiar un tipo de elemento urbano a otro. Por ejemplo, cuando en la actualidad hay un jardín en el que nos gustaría tener un edificio, se necesita un escenario "*what-if*" o variante para saberlo.

Los objetos principales de la ciudad que podrían cambiarse por otro objeto urbano son: los edificios, las vegetaciones, los cuerpos de agua y los transportes, como puede verse en la Figura 3-13.

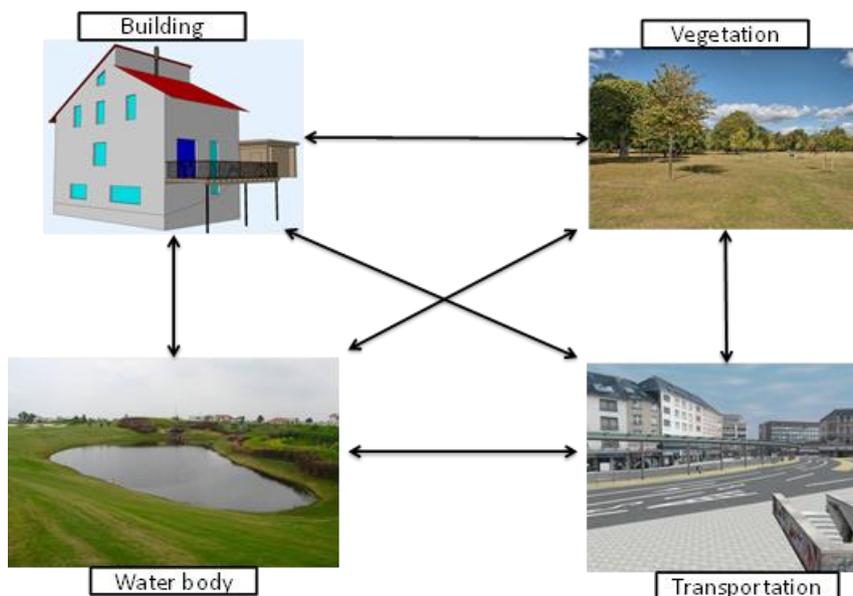


Figura 3-13 Operación de cambiar el tipo de elemento urbano

Para ello, se ha diseñado un esquema XML para expresar los cambios. En este caso se debe usar el siguiente elemento XML 'change'. El tipo original de elemento urbano debe completarse en el atributo 'fromElement' y de la misma manera se indica el nuevo tipo de elemento urbano en el atributo 'toElement'. También se necesita la identificación del elemento urbano original, que se representa en 'fromId'. De esta forma, es posible consultar en el modelo CityGML las características del tipo de elemento urbano original.

```
<change fromElement = "Building" fromId = "1234" toElement = "Vegetation" />
```

Creación / eliminación de nuevo elemento urbano

De esta forma es posible crear y eliminar objetos de ciudad representados como puntos (x, y, z), pero no objetos de ciudad representado por un polígono o una línea. Así es posible crear un nuevo POI representado como CityFurniture (parada de autobús, parada de tranvía, estación de tren, semáforo, parque infantil, etc.).

La creación de un elemento urbano significa la creación de un punto de interés (POI) dentro del módulo CityFurniture. Las siguientes funciones de CityFurniture pueden ser creadas con esta variante: parada de autobús, parada de tranvía, estación de tren, semáforo, parque infantil, etc.

Para hacer eso, se debe usar el elemento 'addPOI'. Luego, el nombre del POI se establece en el atributo 'name' y la información de ubicación se completa con la latitud y la longitud en los atributos 'lat' y 'long'. Finalmente, para establecer la función de CityFurniture, se completa el atributo 'function'.

```
<addPOI name="Parada de autobús - Rua da Ensinanza, 12" lat="42.878323" long="-8.541529" function="1110" />
```

También es posible eliminar algunos de los POI existentes usando la siguiente notación:

```
<removePOI id = "12345" />
```

Cambio de las características de los objetos de la ciudad

Una de las principales mejoras en escenarios *what-if* se logra cambiando alguno de los parámetros semánticos de un elemento urbano. Por ejemplo, un escenario hipotético o variante debería ser poder cambiar la función de un edificio de un edificio industrial a

uno residencial. De esta forma, es posible cambiar las características de los objetos de la ciudad.

En la Tabla 3-2, se presentan algunas las características que pueden ser cambiadas en cada elemento urbano.

Tabla 3-2 Características que pueden ser cambiadas en cada elemento urbano

<p>BUILDING</p> <ul style="list-style-type: none"> • Building Function • Building Type • Area • Noise at façade • Green roof • Number of parking spaces • Year of construction • Number of inhabitants <p>VEGETATION</p> <ul style="list-style-type: none"> • Function • Area <p>TRANSPORTATION</p> <ul style="list-style-type: none"> • Function • Number of parking spaces <p>CITY FURNITURE</p> <ul style="list-style-type: none"> • Function <p>WATER BODY</p> <ul style="list-style-type: none"> • Area <p>DISTRICT</p> <ul style="list-style-type: none"> • Bicycle parking places • Number of parking spaces on street

La variante más utilizada sería el cambio de las características del elemento urbano. Se ha utilizado una notación genérica para permitir cambiar la característica de todos los objetos de la ciudad.

El elemento 'parameter' se debe usar para cambiar las características del elemento urbano. En el atributo 'element' se debe establecer el tipo de objeto y en el 'id' el identificador de la misma. Luego, en 'name' se indica el atributo que debe modificarse y en 'value' el valor deseado.

```
<parameter element="Building" id="54321" name="function" value="1000" />
```

Para cada caso, escenario o variante, se crea un archivo XML con todos los cambios que se realizarían en esa variante.

A continuación, se presenta un ejemplo en el que se contemplan todas las posibles variantes descritas (ver Figura 3-14):

```
<districtVariants>
  <parameter element="District" id="1" name="collectedAndTreatedWater" value="10000" />
  <parameter element="Building" id="2985" name="numberOfInhabitants" value="8" />

  <change fromElement="Building" fromId="2599" toElement="WaterBody" />
  <change fromElement="Building" fromId="2594" toElement="Vegetation" />
  <parameter element="Vegetation" id="2594" name="area" value="300" />
  <parameter element="Vegetation" id="2594" name="function" value="3040" />

  <addPOI name="Parada de autobus - Rua da Ensejanza, 12" lat="42.878323" long="- 8.541529" function="1110" id="123456" />
  <parameter element="CityFurniture" id="123456" name="function" value="1110" />

  <removePOI id = "12345" />
</districtVariants>
```

Figura 3-14 Ejemplo en el que se contemplan todas las posibles variantes

3.4 Implementación

A continuación, se describe el componente “Tecnalia_CityGML_Processing”, que implementa un conjunto de servicios web diseñados para implementar distintos elementos del flujo de trabajo descrito en la sección anterior. Este componente se ha usado con éxito en los proyectos europeos EFFESUS, FASUDIR y HOLISTEEC.

3.4.1 Generación

Las siguientes tablas presentan los detalles de los servicios desarrollados para el *workflow* de generación:

Servicio:	Geometric generation – Create Buildings (LOD0, LOD1 and LOD2)
Descripción:	Servicio de generación masiva del modelo de ciudad en 3D en formato CityGML que contiene edificios en LOD0, LOD1 y LOD2 con información semántica asociada
Entradas:	Varios ficheros SHP con información de edificios, parcelas y zona de interés. Ficheros LIDAR de DSM y DTM.
Proceso:	Se procesan los ficheros LIDAR (tanto DSM como DTM) para obtener la altura media de los edificios. Después se genera cada edificio en LOD0, LOD1 y LOD2 mediante extrusión y ajustándolos al DTM.
Salidas:	Modelo de ciudad en 3D almacenado en una base de datos Postgre con extensión PostGIS
Servicio:	GIS format to CityGML
Descripción:	Servicio de semantización masiva de edificios
Entradas:	Nombre del proyecto en la base de datos; fichero SHP del cual se exporta el .DBF a formato Comma Separated Value (CSV) con la información semántica y fichero XML con las relaciones
Proceso:	El fichero XML asocia los parámetros de CityGML con los datos disponibles en el CSV. Se recorre el fichero CSV y se añade en los diferentes parámetros de CityGML la información contenida en el CSV de forma masiva.
Salidas:	Se almacena en la base de datos de ciudad en 3D la información semántica en el edificio correspondiente

Servicio:	Geometry-based data processing
Descripción:	Se obtienen nuevos parámetros semánticos a partir de los datos geométricos del modelo 3D.
Entradas:	Nombre del proyecto en la base de datos
Proceso:	Utilizando los datos geométricos del modelo 3D se procesan para obtener el área de cada uno de los edificios, y por otro lado, el área, la orientación principal e identificación de las medianeras.
Salidas:	Se almacena en la base de datos la información semántica en los edificios correspondientes

Servicio:	Semantic Processing
Descripción:	Este servicio permite obtener nuevos parámetros semánticos a partir de la información disponible en el modelo CityGML.
Entradas:	Nombre del proyecto en la base de datos. Fichero XML de inferencias.
Proceso:	El fichero XML describe como obtener nuevos parámetros a partir de datos ya existentes en el CityGML. Con lo cual se aplican las formulas descritas en el XML para inferir nuevos parámetros semánticos.
Salidas:	Se actualiza el modelo CityGML con los nuevos parámetros obtenidos.

Servicio:	Crear DB
Descripción:	Este servicio permite crear una base de datos 3DCityDB utilizando PostgreSQL + PostGIS.
Entradas:	Nombre del proyecto y el sistema de referencia a utilizar.
Proceso:	Se procesa el script de generación de la base de datos creando así una nueva base de datos con las tablas y parámetros necesarios para almacenar un fichero CityGML.
Salidas:	Se crea la base de datos espacial que permite almacenar un modelo CityGML

Servicio:	Importar fichero CityGML a la base de datos (BD)
Descripción:	Este servicio permite importar la información del modelo CityGML a la base de datos espacial.
Entradas:	Fichero CityGML y nombre del proyecto en la base de datos.
Proceso:	Se lee la información del fichero CityGML y se almacena toda la información en la base de datos.
Salidas:	Como resultado el fichero CityGML se almacena en una base de datos espacial.

Servicio:	Desplegar WFS
Descripción:	Poner en marcha el servicio WFS asociado a la base de datos CityGML
Entradas:	Nombre del proyecto en la base de datos.
Proceso:	Este servicio permite poner en marcha el servicio WFS que permite acceder al contenido de la base de datos espacial a través de consultas estándares. La herramienta deegree es la encargada de desplegar el WFS.
Salidas:	El WFS desplegado para el proyecto seleccionado.

3.4.2 Operaciones de mantenimiento

Las siguientes tablas presentan los detalles de los servicios desarrollados para las operaciones de mantenimiento:

Servicio:	Unión de ficheros CityGML
Descripción:	Este servicio permite unir dos ficheros CityGML en uno
Entradas:	Dos ficheros CityGML
Proceso:	Este proceso recorre los dos ficheros CityGML y crea un nuevo fichero con la información combinada de ambos ficheros.
Salidas:	Un nuevo CityGML que incluye la información de ambos ficheros CityGML

Servicio:	Reemplazar edificio en CityGML
Descripción:	Reemplaza toda la información de un edificio en concreto
Entradas:	Dos ficheros CityGML. El primero contiene la información del barrio/ciudad mientras que el segundo contiene la información del edificio a reemplazar
Proceso:	Para ello se busca el identificador del edificio disponible en el segundo fichero en el primer fichero CityGML. Después se reemplaza toda la información del edificio
Salidas:	Un nuevo CityGML en el que se ha reemplazado en el primer fichero la información del edificio disponible en el segundo fichero

Servicio:	Servicio de obtener edificios alrededor de un edificio
Descripción:	Obtener los edificios en una distancia X respecto a un edificio concreto
Entradas:	Un fichero CityGML, el identificador de un edificio y la distancia en metros
Proceso:	Para ello se filtran los edificios alrededor del edificio seleccionado a una distancia concreta.
Salidas:	Un nuevo CityGML en el que se han filtrado los edificios que están alrededor del edificio seleccionado a una distancia menor a la indicada

Servicio:	Servicio de exportación de IFC a LOD2/LOD3
Descripción:	Se obtiene, a partir de un fichero IFC, un fichero CityGML con un edificio en LOD2 o LOD3
Entradas:	Un fichero IFC y el LOD de salida deseado
Proceso:	Se exporta tanto la geometría como los datos semánticos del fichero IFC a formato CityGML.
Salidas:	Un fichero CityGML en el que se ha exportado la geometría y semántica del edificio en IFC a CityGML en LOD2 o LOD3

3.5 Aplicación

Como parte del trabajo realizado dentro del proyecto FASUDIR se implementaron tres modelos de ciudad (Santiago de Compostela, Frankfurt y Budapest) siguiendo el *workflow* descrito en la sección 3.3 y utilizando parte de los servicios descritos en la sección 3.4. FASUDIR es un proyecto europeo del 7º Programa Marco desarrollado durante los años 2013-2016. El objetivo del proyecto FASUDIR es desarrollar una herramienta de apoyo a la toma de decisiones (IDST *Integrated Decision Support Tool*) para alcanzar las mejores estrategias de rehabilitación energética urbana para distritos y barrios. El proyecto incorpora el modelo 3D urbano como base de información para analizar el comportamiento energético del entorno edificado en su estado actual y las diferentes alternativas de rehabilitación. El mismo modelo permite la visualización 3D georreferenciada de los principales indicadores de referencia para la toma de decisiones.

Las especificaciones del proyecto han hecho necesario representar más información semántica además de la ya contenida en el estándar CityGML, con lo cual, fue necesario diseñar e implementar unas extensiones de dominio. Como desde el comienzo se ha identificado que las fuentes de datos a la hora de generar cada ciudad eran diferentes, el preprocesado de los datos es independiente para cada ciudad. Sin embargo, el proceso de generación y mantenimiento del modelo CityGML de cada una de las ciudades se ha realizado siguiendo el *workflow* explicado en la sección anterior.

Como la mayoría de las acciones del *workflow* son las mismas para los tres modelos, en los siguientes apartados se detalla cómo se ha realizado la generación del modelo de Santiago de Compostela. Sin embargo, en los casos en los que alguna de las acciones es diferente a la de Santiago de Compostela se describirán dichas diferencias. Describiendo de esta forma cómo se realiza la generación de los tres modelos de ciudades en 3D en CityGML.

3.5.1 Data model set up

(1) Analyze user requirements

La actividad de configuración del modelo de datos incluye como primera acción el análisis de requisitos de usuario. De los requisitos de usuario se han identificado los parámetros extra no contemplados en el estándar CityGML necesarios dentro del proyecto FASUDIR. Para ello se han analizado los requisitos de usuario y se han identificado todos los parámetros necesarios.

Se han definido diferentes parámetros a diferentes escalas, una a escala distrito (áreas de parking, paradas de autobús, metro y tranvía, escuelas, supermercados, etc.) y otra a escala edificio (datos básicos, geometría, combustibles, sistemas, térmico, inversión, etc.). Para cada uno de los parámetros se ha identificado si pertenece o no al CityGML *core*, el tipo de parámetro (numero, texto, fecha, booleano) y fase del *workflow* en la que se incorpora al modelo CityGML.

(2) Search for existing thematic data / Search for existing geometric data

De la misma forma, se ha realizado una búsqueda de datos geométricos y temáticos. Esta es una actividad que consume mucho tiempo, ya que incluye la búsqueda y el preprocesamiento previo de datos de diferentes fuentes, formatos, precisión y fechas, y su posterior combinación.

Un resumen de las fuentes de datos utilizadas para la generación de los modelos CityGML para las tres ciudades del proyecto se presenta en la sección 3.5.2.

En la Tabla 3-3 se presenta un resumen de los datos requeridos para la generación de los modelos 3D urbanos de los tres casos de estudio. La tabla incluye los datos requeridos a nivel de distrito, los datos requeridos para los edificios, así como para los otros elementos de la ciudad (áreas verdes, transporte, mobiliario de la ciudad y cuerpos de agua). Para

cada tipo de datos se identifica la fuente de datos, así como el formato de datos y el alcance de los datos requeridos (solo para el distrito o para el área de caso de estudio completa).

Tabla 3-3 Datos requeridos para generar los modelos 3D urbanos de los casos de estudio

Distrito

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>District Geometry</i>	Proporcionado por Santiago y manualmente editado	Fichero SHP	Distrito	Santiago
<i>District Geometry</i>	Proporcionado por el Instituto Nacional de Estadística y después procesado	Fichero SHP	Otros distritos de la ciudad	Santiago
<i>District Geometry</i>	Proporcionado por Frankfurt y manualmente editado	Fichero SHP	Distrito	Frankfurt
<i>District Geometry</i>	Manualmente generado	Fichero SHP	Otros distritos de la ciudad	Frankfurt
<i>District Geometry</i>	Proporcionado por Budapest y manualmente editado	Fichero SHP	Distrito	Budapest
<i>District Geometry</i>	Manualmente generado	Fichero SHP	Otros distritos de la ciudad	Budapest

Building

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>Building and Building</i> LOD0, LOD1, LOD2	Proporcionado por Santiago y después procesado	Fichero SHP	Distrito	Santiago
<i>Building and Building</i> LOD0, LOD1, LOD2	Proporcionado por Catastro España y después procesado	Fichero SHP	A 2 km de radio alrededor	Santiago
<i>Digital Model (DSM)</i> Surface	Proporcionado por el Instituto Geográfico Nacional con 1m de resolución. Procesado para exportar de LAS a ASC	Fichero ASC	A 2 km de radio alrededor	Santiago
<i>Digital Model (DTM)</i> Terrain	Proporcionado por el Instituto Geográfico Nacional con 5m de resolución. Procesado para exportar de LAS a ASC	Fichero ASC	A 2 km de radio alrededor	Santiago
Building and Building LOD0, LOD1, LOD2	OSM y después procesado	Fichero SHP	Distrito	Frankfurt
Building and Building LOD0, LOD1, LOD2	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Frankfurt
<i>Digital Model (DSM)</i> Surface	Proporcionado por Frankfurt con 1m de resolución. Procesado para exportar de XYZ a ASC	Fichero ASC	A 2 km de radio alrededor	Frankfurt
<i>Digital Model (DTM)</i> Terrain	Proporcionado por Frankfurt con 1m de resolución. Procesado para exportar de XYZ a ASC	Fichero ASC	A 2 km de radio alrededor	Frankfurt
Building and Building LOD0, LOD1, LOD2	Proporcionado por Budapest en formato AutoCAD DXF y después procesado	Fichero SHP	Distrito	Budapest

Building and Building	LOD0 LOD1, LOD2	Proporcionado por Budapest en formato AutoCAD DXF y después procesado	Fichero SHP	A 2 km de radio alrededor	Budapest
<i>Digital Model (DSM)</i>	<i>Surface</i>	No disponible			Budapest
<i>Digital Model (DTM)</i>	<i>Terrain</i>	Proporcionado por Budapest con 5m de resolución. Procesado para exportar de XYZ a ASC	Fichero ASC	A 2 km de radio alrededor	Budapest

Building – Facilities and Services

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>Schools, Kindergarten, Supermarket, Health facility, Bank, Restaurant, Bar, Indoor sport facility, Outdoor sport facility</i>	Proporcionado por Santiago y después procesado	Fichero SHP	A 2 km de radio alrededor	Santiago
Schools, Kindergarten, Supermarket, Health facility, Bank, Restaurant, Bar, Indoor sport facility, Outdoor sport facilities	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Frankfurt
<i>Schools, Kindergarten, Supermarket, Health facility, Bank, Restaurant, Bar, Indoor sport facility, Outdoor sport facilities</i>	Proporcionado por Budapest en formato AutoCAD DXF y después procesado	Fichero SHP	A 2 km de radio alrededor	Budapest

Green áreas

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>Green dense population, Green light population, Small gardens and green backyards</i>	Proporcionado por Santiago y después procesado	Fichero SHP	Distrito	Santiago
<i>Grass field, Forest area</i>	OpenStreetMap (grass fields) OSM a SHP	Fichero SHP	A 2 km de radio alrededor	Santiago
<i>Green dense population; Green light population; Small gardens and green backyards; Grass field; Forest area</i>	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Frankfurt
<i>Green dense population, Grass field</i>	Proporcionado por Budapest en formato AutoCAD DXF y después procesado	Fichero SHP	A 2 km de radio alrededor	Budapest
<i>Green light population; Small gardens and green backyards; Forest area</i>	No disponible			Budapest

Transportation

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>Railway, Roads</i>	Proporcionado por Santiago y después procesado	Fichero SHP	A 2 km de radio alrededor	Santiago
<i>Square with green elements, Parking Areas, Number of parking spaces in each parking area</i>	Proporcionado por Santiago y después procesado	Fichero SHP	Distrito	Santiago
<i>Railway, Roads, Square with green elements, Parking Areas</i>	OSM y después procesado	Fichero SHP	Distrito	Frankfurt
<i>Railway, Roads</i>	OSM y después procesado	Fichero SHP	Distrito	Budapest
<i>Square with green elements</i>	Proporcionado por Budapest en formato AutoCAD DXF y después procesado	Fichero SHP	A 2 km de radio alrededor	Budapest
<i>Parking Areas; Number of parking spaces in each parking area</i>	No disponible			Budapest

City Furniture

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>Bus stops,</i> <i>Railway stops,</i> <i>Signal light,</i> <i>Playground,</i> <i>Public open space</i>	Proporcionado por Santiago y después procesado	Fichero SHP	A 2 km de radio alrededor	Santiago
<i>Tram stops,</i> <i>Metro stops</i>	No aplica en Santiago	---	---	Santiago
Bus stops, Tram stops, Railway stops, Metro stops, Signal light, Playground, Public open space	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Frankfurt
Bus stops, Tram stops, Signal light	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Budapest
Railway stops; Metro stops, Playground; Public open space	No disponible			Budapest

Water Bodies

DATOS REQUERIDOS	FUENTE DE DATOS	FORMATO DE DATOS	AMBITO	CIUDAD
<i>Lakes, pounds and rivers</i>	Proporcionado por Santiago y después procesado	Fichero SHP	A 2 km de radio alrededor	Santiago
Lakes, pounds and rivers	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Frankfurt
Lakes, pounds and rivers	OSM y después procesado	Fichero SHP	A 2 km de radio alrededor	Budapest

(3) Data pre-processing

En el caso del catastro de España, se dispone de información a dos escalas: parcelas y unidades constructivas. Las parcelas contienen la geometría de toda la parcela (dentro de la cual pueden existir varios edificios) y dispone de información semántica como: año de construcción, uso principal o número de plantas. Por otro lado, las unidades constructivas permiten disponer de una geometría más detallada y cercana al del edificio; pero de la que se dispone de poca/ninguna información semántica. Esto hace necesario el preprocesado de esta información con el objetivo de adecuar la geometría (seleccionando la más realista y detallada, la de las unidades constructivas); realizar una limpieza de datos (desechando los datos no relevantes) y organizar los datos (uniendo la información semántica de las dos escalas en una).

La Figura 3-15 detalla el proceso realizado para este preprocesado y se realiza utilizando una herramienta GIS de escritorio. El polígono con rallas representa la parcela (la cual tiene la mayor información semántica) y los polígonos con puntos los edificios (de la que nos interesa la geometría) dentro de esa parcela. El primer paso consiste en seleccionar la capa de edificios y obtener un punto centroide de cada uno de ellos, este punto incluye la información semántica del edificio. Después se realiza la intersección de la capa de centroides del edificio con la capa de parcelas, obteniendo así una capa de puntos

(centroides) en la que para cada punto se tiene la información semántica del edificio y también la de la parcela con la que se ha intersecado. Una vez que tenemos esto, se realiza la unión de la información semántica de estos puntos con la capa de edificios. Obteniendo así la mayor cantidad de información semántica a escala edificio.

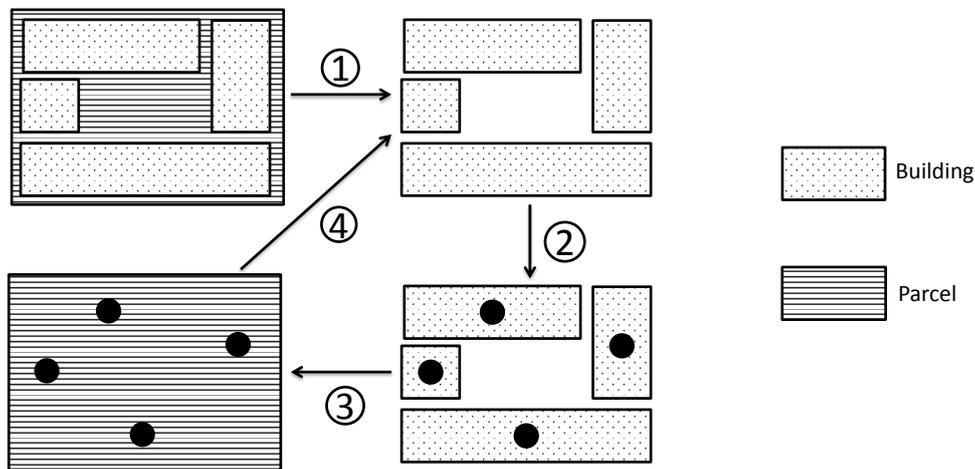


Figura 3-15 Proceso realizado para realizar el preprocesado entre parcelas y edificios

(4) Application domain extension required

A partir de la identificación de los requisitos de usuario, se han diseñado y generado tres extensiones (a escala edificio, a escala distrito y otros elementos) utilizando las ADE de CityGML; para así poder representar los parámetros necesarios dentro de FASUDIR.

ADE a escala edificio

La Figura 3-16 ilustra el resumen de las ADE requeridas en la escala de edificio. En el módulo de edificio se presentan los parámetros de la especificación CityGML que se usan en FASUDIR. Alrededor del módulo de edificio se presentan los nuevos objetos necesarios a escala de edificio: *BuildingGeometry*, *BuildingProperties*, *Interventions*, *Investment*, *Fuels*, *Systems*, *Thermal*, *Environment* y *Building Key Performance Indicator (KPI)*.

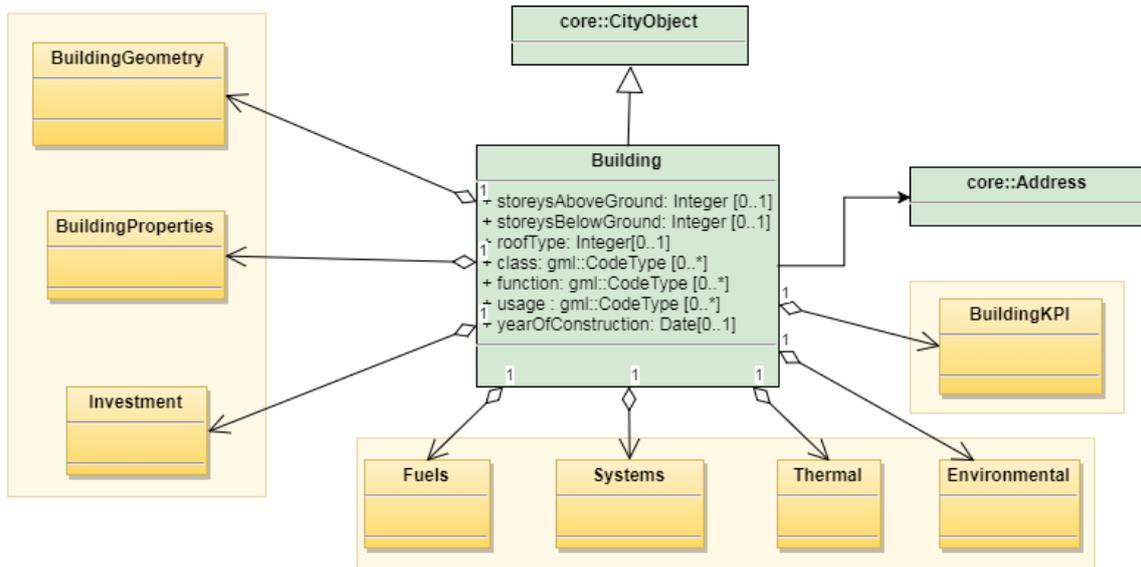


Figura 3-16 ADE a escala edificio

Building general

En la Figura 3-17 se ilustran las ADE creadas con información general a escala de edificio. Con respecto a BuildingGeometry se presenta el porcentaje de área de ventanas en diferentes fachadas, así como datos del tejado. Dentro de BuildingProperties se almacenan los datos generales del edificio, como la condición del edificio o el número de habitantes. También se almacena información sobre intervenciones pasadas (Interventions). Finalmente, se identifican los datos de inversión para el edificio (Investment).

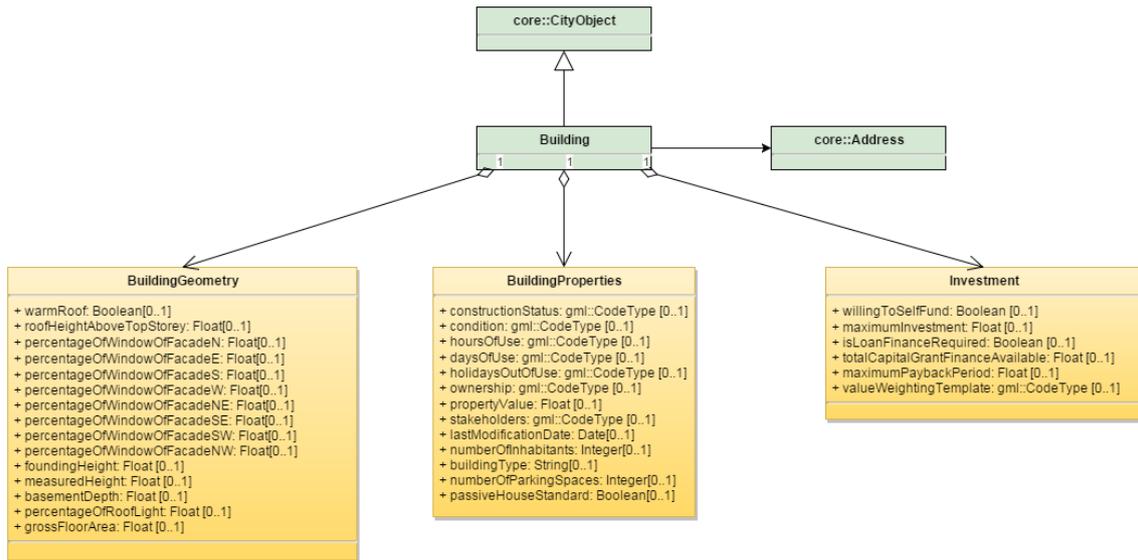


Figura 3-17 ADE building general

Building energy

En la Figura 3-18 se ilustran las ADE relacionadas con la energía a escala edificio. Dentro de la building energy ADE se identifica información sobre el uso de combustibles en el edificio, como calefacción y refrigeración (Fuels). También se representan los datos sobre el sistema instalado para calefacción, ventilación y aire acondicionado (Systems). En cuanto a las propiedades térmicas, se almacenan también los materiales utilizados en paredes, techos, etc. (Thermal). Finalmente, se incluyen ciertos datos ambientales (Environmental).

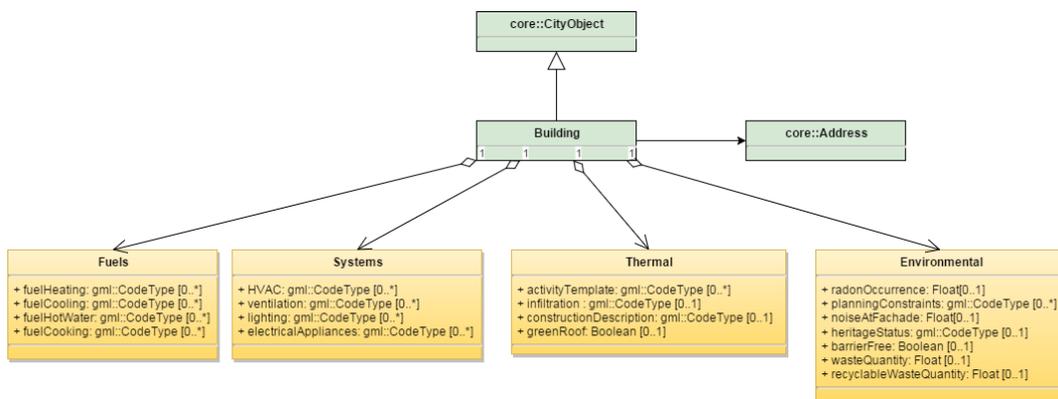


Figura 3-18 ADE building energy

Building KPI

El building KPI ADE incluye todos los KPI identificados en la escala de edificio (ver Figura 3-19). Dentro de cada proyecto es posible crear diferentes escenarios, y dentro de cada escenario se pueden probar diferentes variantes; por lo cual, se asigna un número de identificación para el proyecto, escenario y variante para cada cálculo KPI del edificio. Permitiendo así el almacenamiento de los KPI de edificios resultado de cada variante, como se ha explicado previamente en la sección 3.3.2.

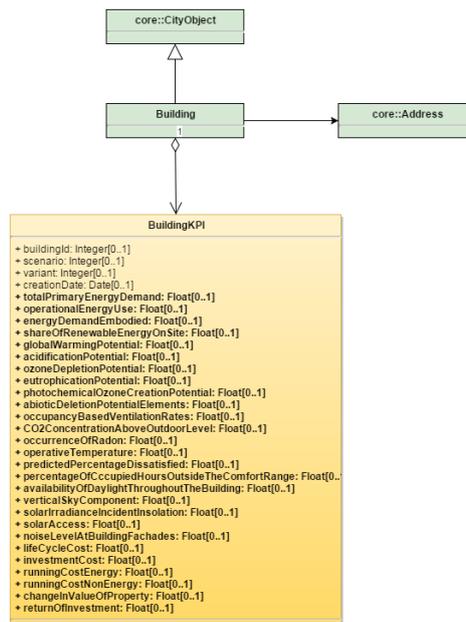


Figura 3-19 ADE building KPI

ADE a escala distrito

En la Figura 3-20 se presenta el resumen de las ADE a escala de distrito. Alrededor del módulo city object group, que se utiliza en FASUDIR para representar los distritos, se definen los siguientes objetos a escala de distrito: el uso de tierras, elementos de agua, transporte e infraestructuras. También se almacenan datos de accesibilidad, confort térmico, gentrificación y nivel de ruido. Finalmente, los costos del ciclo de vida y el retorno de los datos de inversión también están representados a escala distrito.

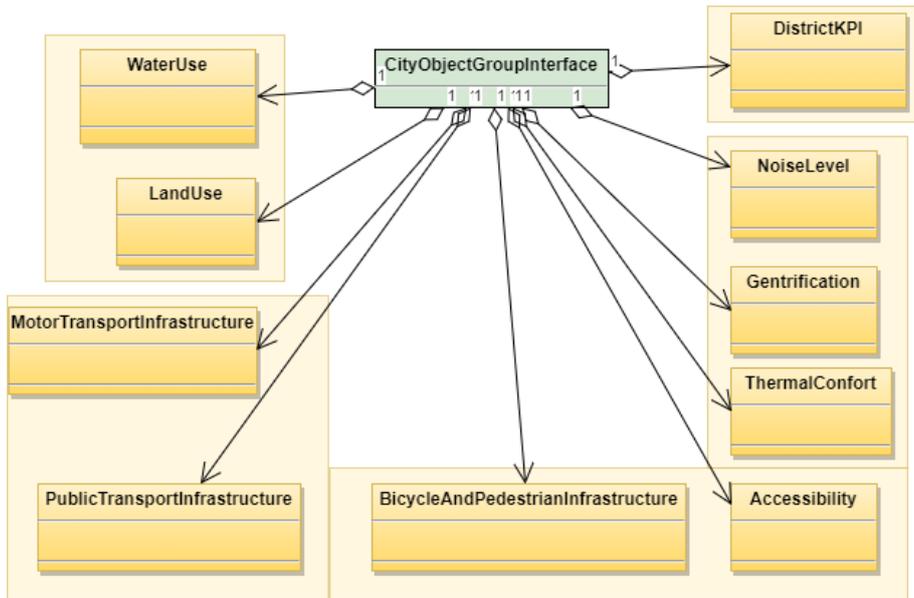


Figura 3-20 ADE a escala distrito

District environmental

En la Figura 3-21 se ilustra la ADE de categoría medioambientales a escala de distrito. Incluyen la información requerida para el cálculo de los KPI medioambientales, que se divide en información sobre el uso del agua (WaterUse) y el uso del suelo (LandUse).

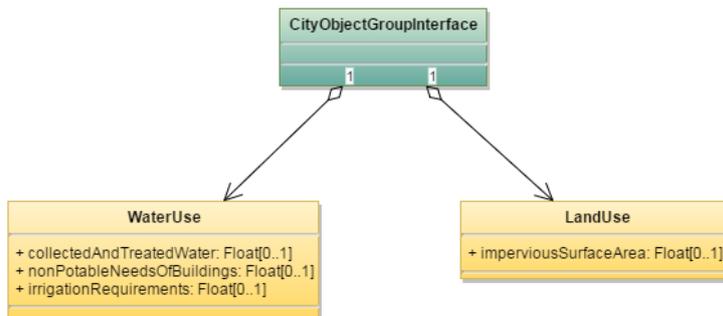


Figura 3-21 ADE district environmental

District social

En la Figura 3-22 se presenta la ADE de categoría social a escala de distrito e incluye la información requerida para el cálculo de KPI sociales. Dentro de esta ADE se presenta información de la infraestructura de transporte con parámetros como el número de espacios de estacionamiento (MotorTransportationInfrastructure). También se almacenan los datos de las infraestructuras de transporte público (PublicTransportInfrastructure) y las infraestructuras de bicicletas y peatones (BicycleAndPedestrianInfrastructure). También se identifica información acerca de la accesibilidad del distrito, como el área de los edificios que se encuentra a cierta distancia de un punto de interés (Accessibility). Otro módulo es el confort térmico a nivel de distrito, donde se presenta información de tipologías de edificios y área ocupada por zonas de vegetación (ThermalConfort). Finalmente, se identifica un módulo para medir el nivel de ruido a nivel de distrito (NoiseLevel) e información sobre gentrificación (Gentrification).

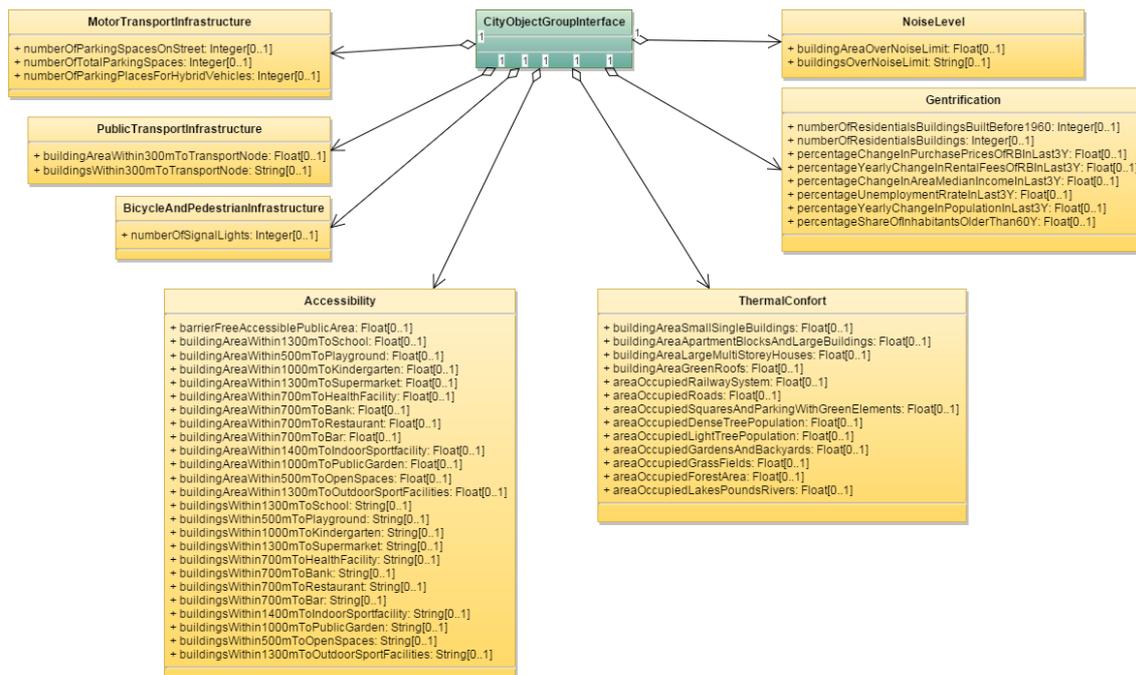


Figura 3-22 ADE district social

District KPI

La ADE de District KPI incluye todos los KPI identificados a escala de distrito (ver Figura 3-23). Dentro de cada proyecto es posible crear diferentes escenarios, y dentro de cada escenario se pueden probar diferentes variantes; por lo cual, se asigna un número de identificación para el proyecto, escenario y variante para cada cálculo KPI del distrito. Permitiendo así el almacenamiento de los KPI de distritos resultado de cada variante.



Figura 3-23 ADE district KPI

ADE de otros elementos

La ADE de otros elementos incluyen las extensiones requeridas para todos los objetos de la ciudad a escala de distrito (Buildings, CityFurniture, Transportation, Vegetation y WaterBody) (ver Figura 3-24). Se añade un atributo en el edificio para la medición del nivel de ruido. Se representa la información general del distrito, como nombre, país, área total o número de habitantes. Para el transporte, las zonas verdes y los objetos de agua también se incluye información sobre el área de cada elemento.

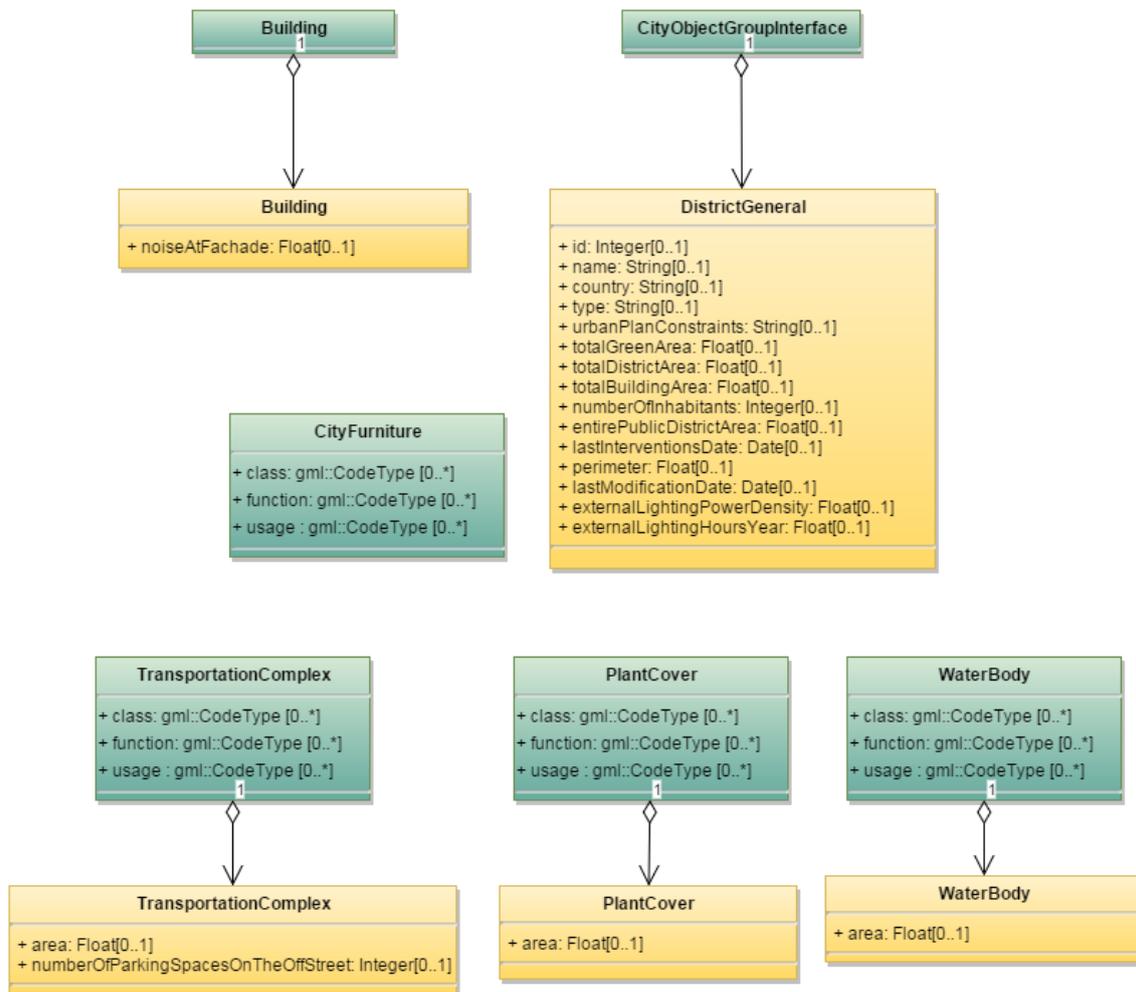


Figura 3-24 ADE de otros elementos

3.5.2 Geometric generation

(5) Search for available CityGML 3D models

Como no existe ningún modelo CityGML disponible de Santiago de Compostela y de Budapest, se ha realizado la generación de los mismos. Por otro lado, aunque ya existe un modelo CityGML de Frankfurt, el nivel de detalle geométrico no está a escala edificio y tampoco dispone de información semántica asociada. Tampoco es fácil asociar la información semántica ya que los identificadores no coinciden. Con lo cual, se ha decidido generar el modelo CityGML utilizando el *workflow* descrito y utilizando datos de OSM.

(6) Create Buildings (LOD0 LOD1 and LOD2)

El casco antiguo de Santiago es Patrimonio de la Humanidad y, por lo tanto, está protegido por severas restricciones de preservación histórica y está compuesto mayoritariamente de edificio terciario y residencial (ver Figura 3-25).

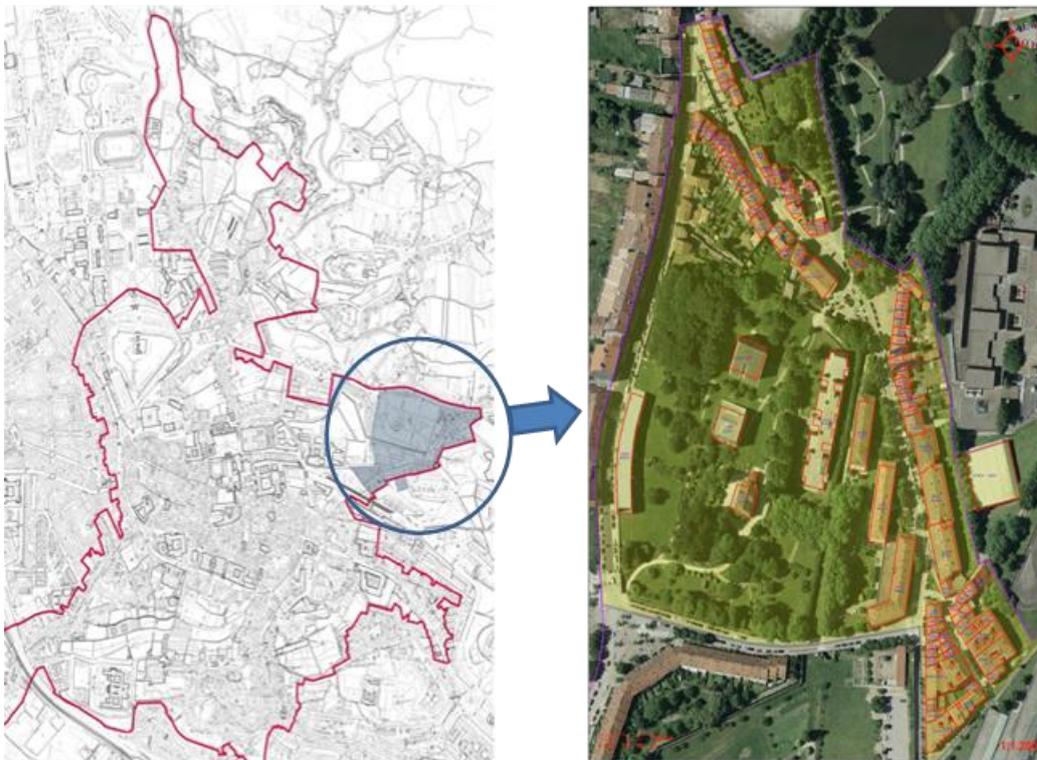


Figura 3-25 Distrito de Santiago de Compostela

El distrito a analizar en FASUDIR se ubica en la parte noroeste del área incluida en la Lista del Patrimonio Mundial de la UNESCO, específicamente en un tejido urbano "histórico lineal" que se conecta con el núcleo central (almendra) de la Ciudad Vieja en

la Puerta de San Francisco. El número total de edificios del distrito es 92, pero se clasifican por tipología según el año de construcción, densidad (número de viviendas por edificio), técnicas de construcción, etc. Estos últimos bloques tienen una baja calidad constructiva y poca protección frente al clima.

La geometría de los edificios se representa con diferentes niveles de detalle (LOD0, LOD1 y LOD2). Para otros elementos de la ciudad solo se genera LOD1.

Edificios

En el caso de Santiago de Compostela los edificios de interés son los de la Figura 3-26. Sin embargo, con el objetivo de poder calcular todos los KPI del proyecto se ha decidido generar el modelo de un área de radio de 2 km alrededor del área de interés (ver Figura 3-27). Solo se han completado los datos semánticos del área de interés. El modelo adicional se crea para realizar los cálculos necesarios para evaluar las distancias entre diferentes objetos de la ciudad en los KPI de FASUDIR.

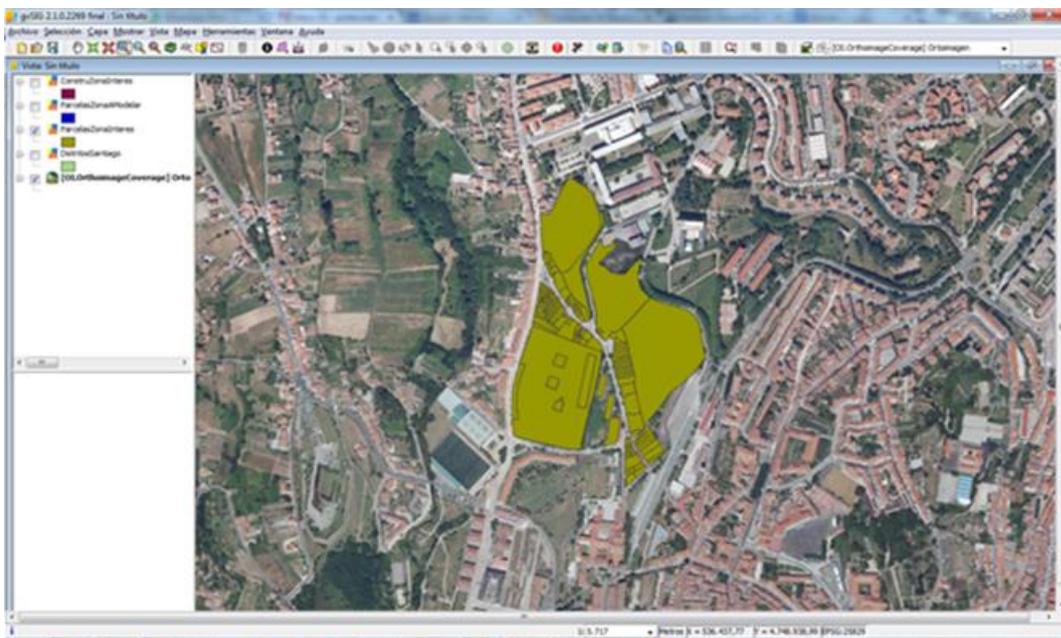


Figura 3-26 Zona de interés de Santiago de Compostela

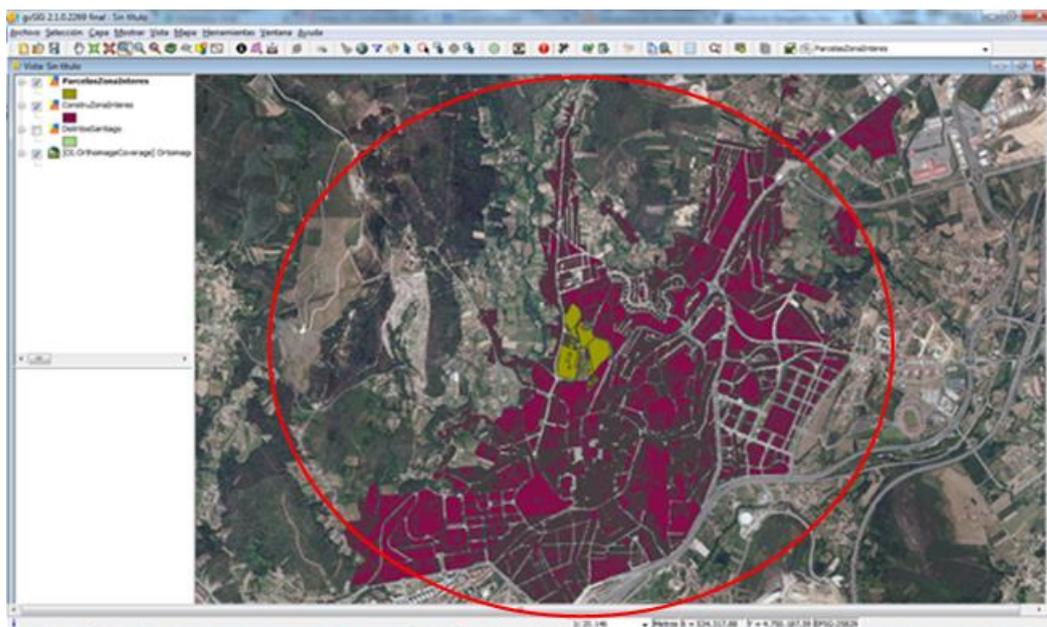


Figura 3-27 Generar el modelo de un área de radio de 2 km alrededor del área de interés

En la herramienta gvSIG se selecciona EPSG: 25829 como sistema de referencia para Santiago de Compostela. La capa de distritos se obtiene del Instituto Nacional de Estadísticas y la capa CONSTRU (con 27.301 registros) de la Dirección General de Catastro. Después se han realizado varios preprocesos geográficos para limpiar y obtener los datos relevantes de cada capa.

El siguiente paso es obtener la altura media de todos los edificios de la zona de estudio. Para ello se han procesado los ficheros LIDAR, tanto los DSM como los DTM, obteniendo así una altura media para cada edificio.

Después, se procede a generar los edificios en LOD0, LOD1 y LOD2. El LOD0 se obtiene utilizando la planta de la geometría del edificio asociándolo al DTM. Sin embargo, para el LOD1 y LOD2 es necesario utilizar la altura media del edificio para extraerlo. En el caso de LOD2 es necesario diferenciar semánticamente que superficies son tejados y cuales fachadas.

(7) Add Other City Elements (green areas, roads, POIs, 3D models)

Transporte / Mobiliario urbano / Vegetación / Zonas de agua

Se ha realizado la generación del resto de elementos de la ciudad en LOD1, en la que se incluyen: elementos de transporte, mobiliario urbano, vegetación y zonas de agua del modelo de ciudad de Santiago.

Para esta generación se han utilizado los datos identificados en la sección 3.5.1.

Datos OSM

En la Tabla 3-4 se detalla la relación entre los tipos de datos de FASUDIR y los datos contenidos en OSM que después han sido utilizado en el caso de Frankfurt y Budapest.

Tabla 3-4 Información utilizada para la generación de los elementos de transporte, mobiliario urbano, vegetación y zonas de agua del modelo de ciudad de Frankfurt

		OSM				
FASUDIR TYPE	Building	Natural	Points	Railway	Road	Waterway
TRANSPORTATION	Parking Areas	parking	motorcycle_parki, parking, parking_entrance	abandoned, disused, light_rail, platform, rail, subway, tram		
	Railway System		railway_crossing		bridleway, cycleway, footway, living_street, motorway, motorway_link, path, pedestrian, primary, primary_link, proposed, residential, secondary, secondary_link, service, services, steps, tertiary, track, unclassified	
	Road / Street		motorway_junctio			
	Square with green elements					
CITYFURNITURE	Bus Stop		bus_stop			
	Tram Stop		tram_stop			
	Metro Stop		subway_entrance			
	Railway Station		station			
	Signal Light		traffic_signals			
	Playground					
	Public Open Space					
Outdoor Sport Facility			sports			
VEGETATION	Public Garden		park			
	Green space, dense tree population					
	Green Space, light tree population					
	Small gardens and green backyards					
	Grass field					
Forest area		forest				
WATERBODY	Lakes, ponds and rivers		water, riverbank			ditch, river, stream, weir

Resultados Santiago

Como resultado de la generación geométrica de los edificios del modelo 3D urbano para el caso de estudio de Santiago, se han generado edificios con diferentes niveles de detalle (LOD0, LOD1 y LOD2) para el área de caso de estudio y radio de 2km alrededor del distrito.

El caso de estudio se compone de 92 edificios, sin embargo, como se ha generado 2km alrededor del caso de estudio, se han generado 6.525 edificios en total. Se han generado también: 8 elementos de transporte, 197 elementos de vegetación y 809 puntos de mobiliario urbano (ver Figura 3-28).

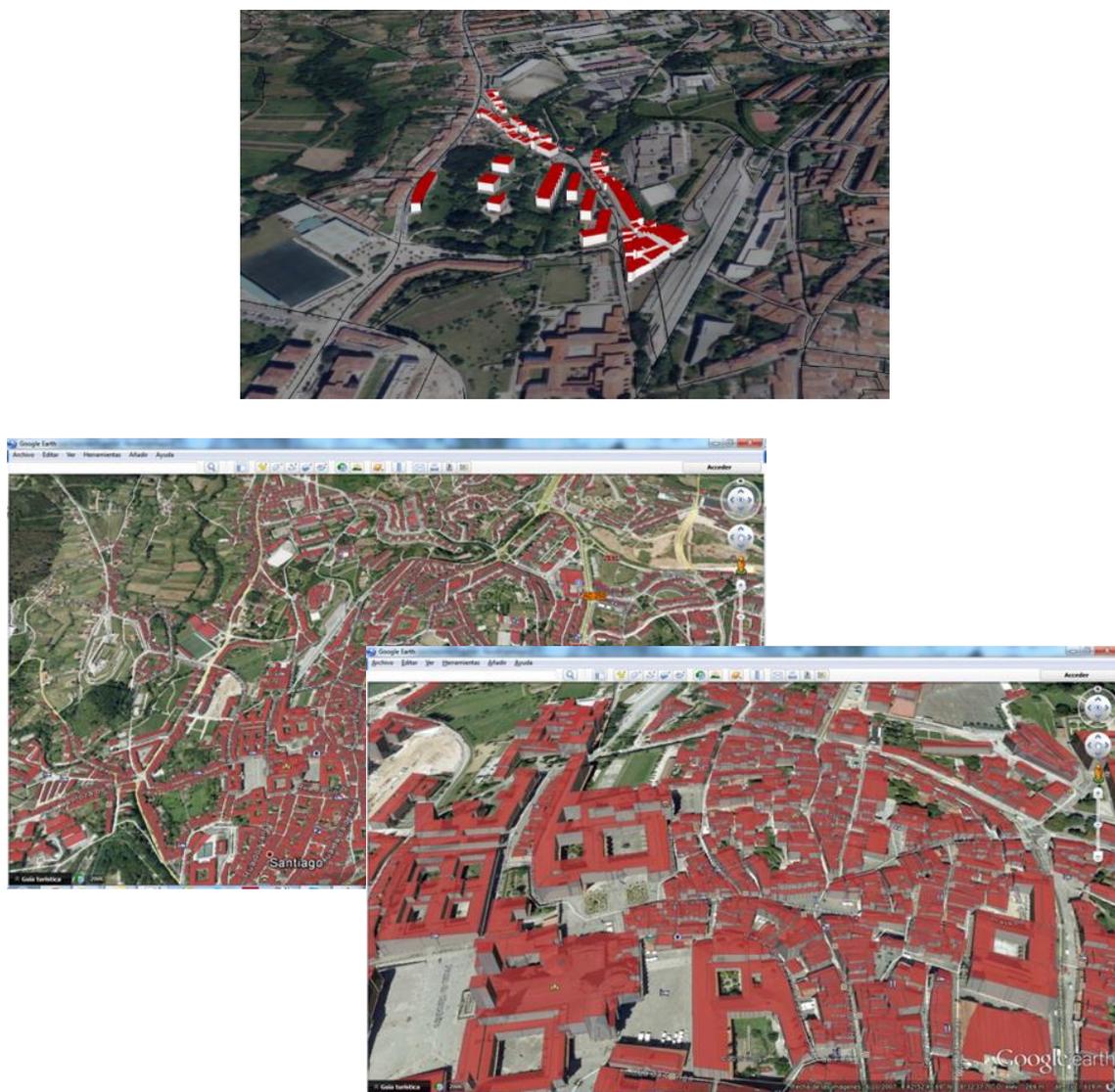


Figura 3-28 Resultado de Santiago de Compostela en CityGML

Resultados Frankfurt

A continuación, se presentan unas ilustraciones de la zona de estudio del caso de Frankfurt y el resultado obtenido en CityGML. El caso de estudio se compone de 45 edificios, sin embargo, como se ha generado 2km alrededor del caso de estudio, se han generado 6.460 edificios en total. Se han generado también: 154 elementos de transporte, 23 elementos de vegetación, 162 puntos de mobiliario urbano y 9 objetos de agua (ver Figura 3-29). En la Figura 3-30 se presenta el resultado en CityGML.



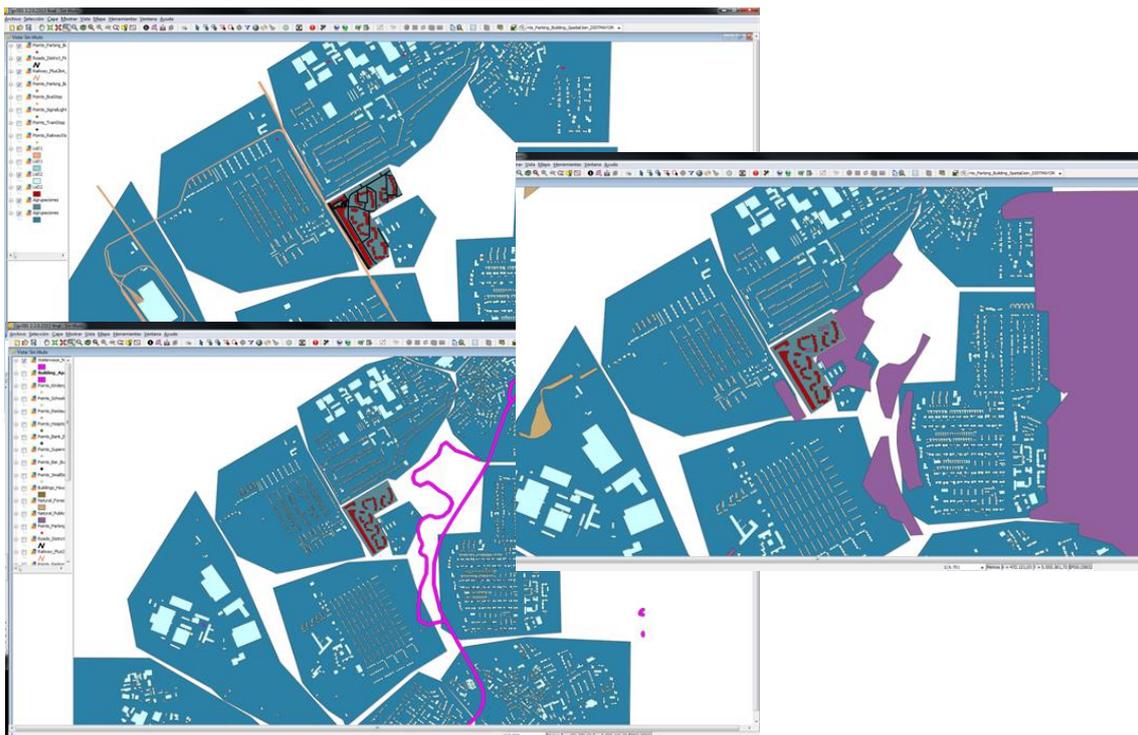


Figura 3-29 Información en 2D de Frankfurt

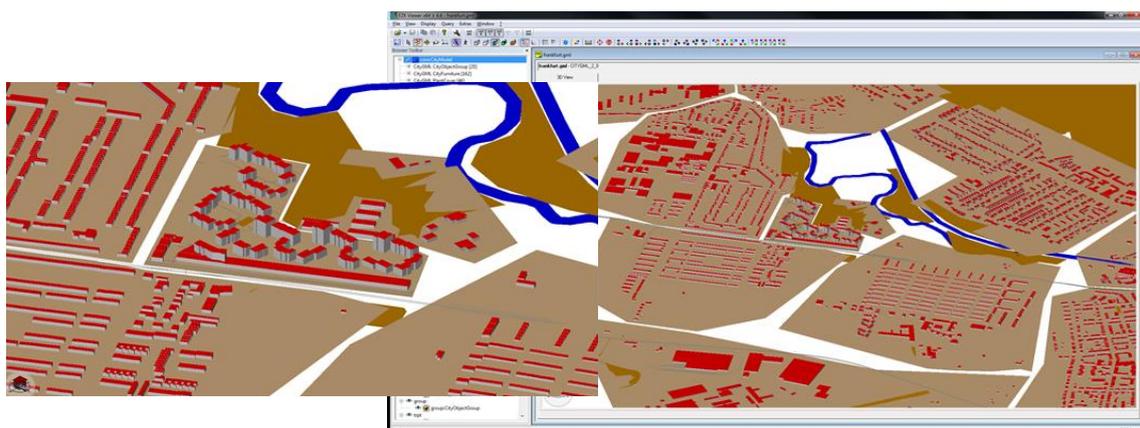


Figura 3-30 Resultado de Frankfurt en CityGML

Resultados Budapest

A continuación, en la Figura 3-31 se ilustra la zona de estudio del caso de Budapest.

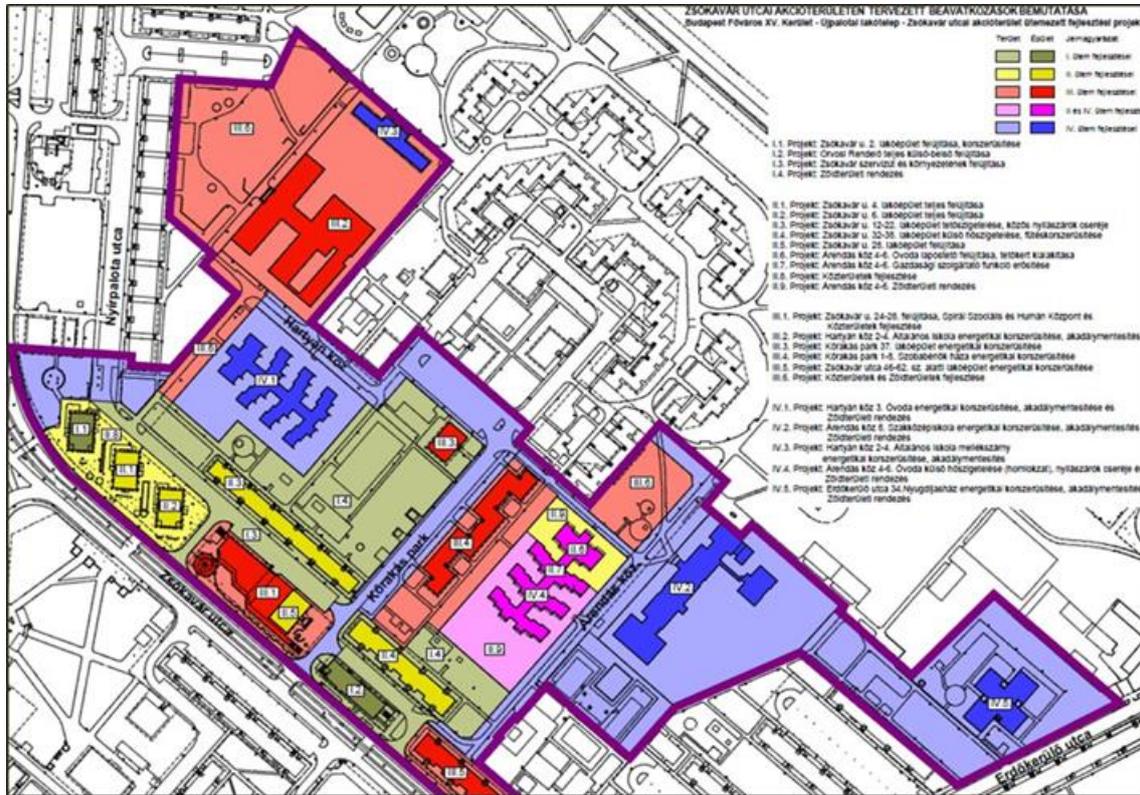


Figura 3-31 Zona de interés de Budapest

Como resultado de la Generación Geométrica de los edificios en el Modelo de Ciudad para el caso de estudio de Budapest, se han generado edificios con diferentes niveles de detalle (LOD0, LOD1 y LOD2) para el área de caso de estudio y radio de 2km alrededor del distrito. El caso de estudio se compone de 19 edificios, sin embargo, como se ha generado 2km alrededor del caso de estudio, se han generado 3.074 edificios en total. Se han generado también: 31 elementos de transporte y 111 puntos de mobiliario urbano (ver Figura 3-32).

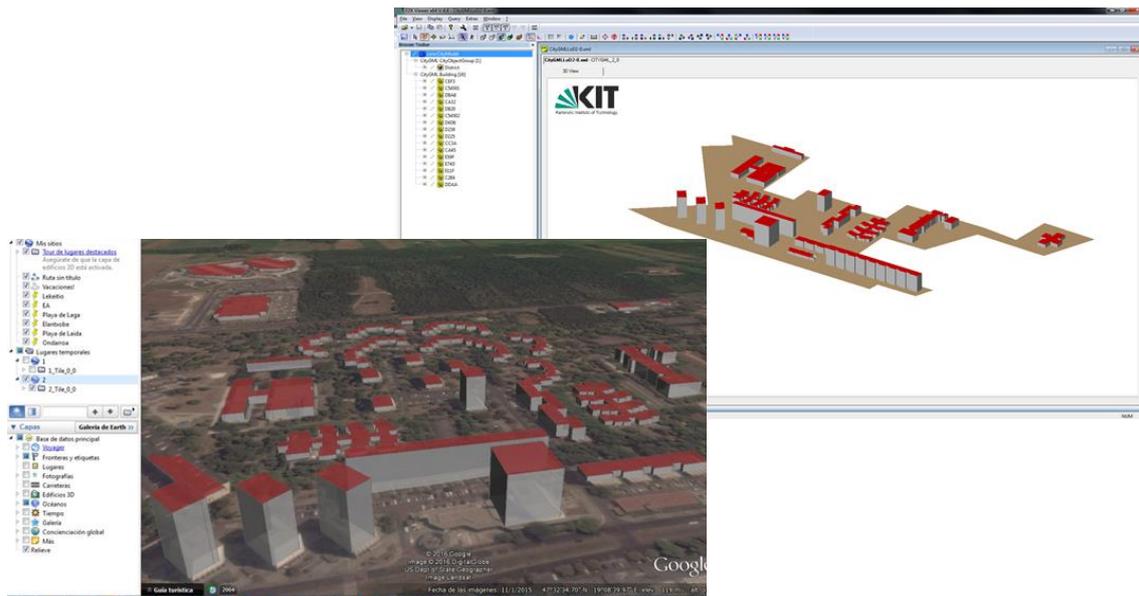


Figura 3-32 Resultado de Budapest en CityGML

(8) Insert into DB

Los tres casos de estudios han sido importados a la base de datos espacial 3DCityDB que utiliza PostgreSQL + PostGIS. Además, se han generado los scripts para añadir los nuevos parámetros de las ADE en la base de datos. Del mismo modo, se ha desplegado el WFS para los tres casos de estudio, para así poder acceder a los datos realizando peticiones estándares.

3.5.3 Semantic generation

La generación semántica permite semantizar el modelo CityGML de diferentes formas: utilizando información recopilada, procesando la información geométrica o mediante la inferencia de nuevos datos semánticos.

(9) GIS format to CityGML

Semantizacion de los edificios en el área del distrito

Este proceso consiste en completar los datos de los edificios en el área de estudio. Para ello se utiliza, por un lado, un fichero SHP del cual se exporta el .DBF a formato CSV, que contiene la información semántica. Completar este fichero CSV implica tener que recopilar datos de diferentes fuentes de datos. Por otro lado, se genera un fichero XML en el que se indican las relaciones entre el CSV y los parámetros en CityGML. De esta

forma es posible mapear a que parámetro de CityGML se asocia cada columna del CSV. Después, recorriendo el fichero CSV se completa la información del CityGML, consiguiendo así completar la información semántica de forma masiva.

(10) Geometry-based data processing

A continuación, se detallan los procesos geométricos realizados. Para ello se procesa toda la geometría de los edificios del modelo CityGML y se identifican las medianeras y se calculan la orientación principal de las fachadas y medianeras.

Medianeras

A continuación, se presenta el resultado del procesado geométrico obteniendo las fachadas que son medianeras de la zona de estudio (ver Figura 3-33):

Leyenda: Amarillo – Fachada Exterior y Rojo – Fachada medianera

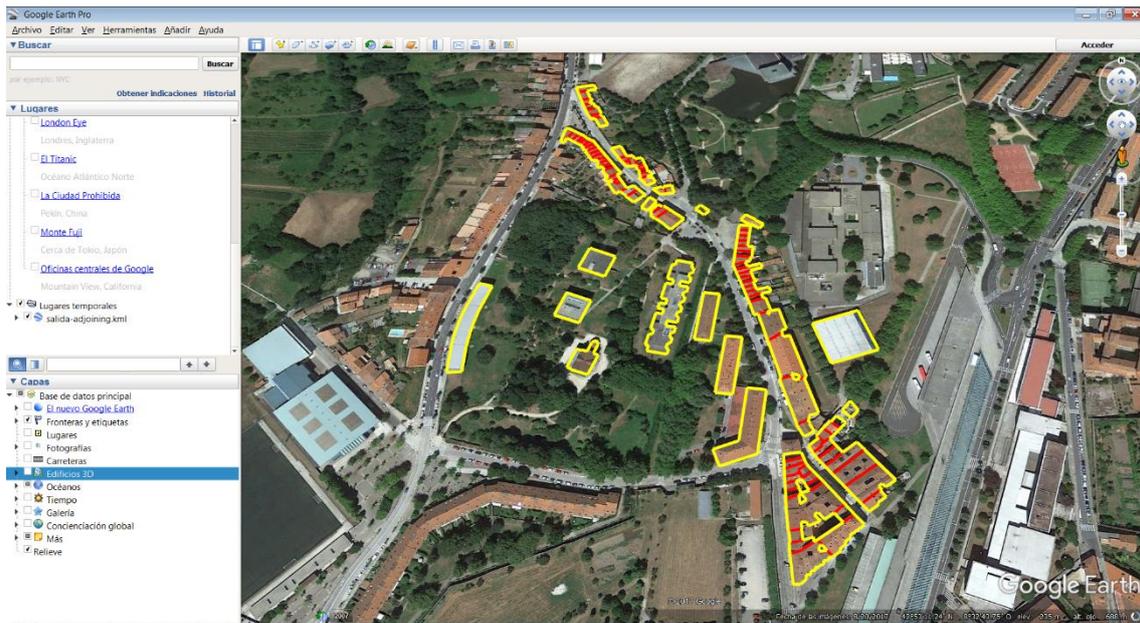


Figura 3-33 Procesado geométrico de medianeras

Orientación principal

A continuación, se presenta el resultado del procesado geométrico obteniendo la orientación principal de cada una de las fachadas de la zona de estudio (ver Figura 3-34):

Leyenda: Blanco – Oeste; Rojo – Norte; Verde – Este y Azul - Sur

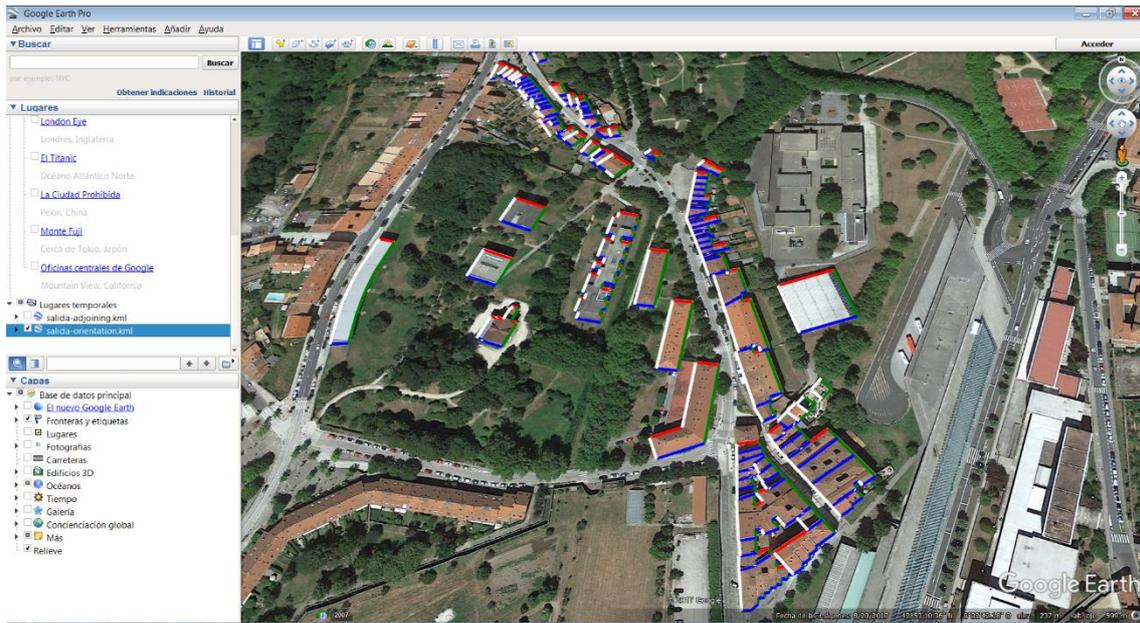


Figura 3-34 Procesado geométrico de la orientación principal

Otros procesos geométricos llevados a cabo en los modelos del proyecto FASUDIR han permitido completar el modelo con información de volumen y áreas del edificio.

(11) Semantic processing

Se realizan dos tipos de inferencias. La primera, obteniendo nueva información semántica en base a otros parámetros. Aplicando formulas o condiciones. Y la segunda mediante sumatorios o agregación de datos. Por ejemplo, el número de habitantes puede agregarse a escala distrito sumando los habitantes de cada edificio.

Un ejemplo de obtener un nuevo parámetro a través de la inferencia consiste en obtener un periodo a través del año de construcción. Las herramientas de simulación necesitan como entrada el periodo construcción en vez del año de construcción, ya que así pueden aplicar plantillas predefinidas.

(12) Check if data is complete

La última acción es asegurarse de que no se necesita ninguna otra información. En el proyecto FASUDIR la herramienta de toma de decisiones resultante permite la visualización y edición de las propiedades de los elementos urbanos de los modelos CityGML de las tres ciudades.

3.5.4 Operaciones de mantenimiento

Se han implementado diferentes operaciones de mantenimiento dependiendo del proyecto en el que se ha trabajado.

En el proyecto U3DCloud se generó un modelo CityGML en LOD2 de un barrio de Vitoria - Gasteiz y se completó la información semántica de los edificios con los servicios previamente descritos. Sin embargo, al de un tiempo, se ha identificado la necesidad de tener que ampliar el modelo CityGML. Con lo cual, en vez de generar todo de nuevo, se ha generado otro barrio en LOD2 y utilizando el servicio de concatenar varios modelos CityGML se ha unificado en un único fichero. Se ha identificado también que en Google 3D Warehouse existían modelos geoméricamente más detallados y con texturas de algunos edificios de la zona. Con lo cual, se ha realizado la exportación de esos edificios a CityGML. A la hora de realizar el reemplazo del nuevo edificio en CityGML con el edificio antiguo, en vez de hacerlo a mano se ha utilizado el servicio reemplazar un edificio por otro en un modelo CityGML. Estos servicios, aunque son sencillos, se utilizan muy a menudo y evitan tener que realizarlo manualmente, reduciendo así también los posibles errores.

En el proyecto HOLISTEEC, el objetivo es realizar la simulación de la eficiencia energética utilizando EnergyPlus³⁶ a partir de un IFC. Sin embargo, con el objetivo de realizar una simulación más realista, es necesario disponer de los edificios cercanos al edificio a simular, es decir, disponer del entorno en CityGML. Para ello, en vez de utilizar todo el fichero CityGML de la zona de interés, se filtra el CityGML para obtener únicamente los edificios a una distancia determinada (50 o 100m) alrededor de un edificio en concreto. Este proceso hay que repetirlo por cada edificio que se quiere simular.

En el mismo proyecto se ha identificado la necesidad de incluir el fichero IFC (que era un edificio no disponible en el modelo CityGML, ya que la última versión del catastro no estaba representado) en CityGML. Para ello se utilizó el servicio completar el modelo CityGML con un modelo de un edificio más detallado. Este complejo servicio mapea todos los objetos de IFC a los correspondientes en CityGML y transforma la geometría de un sistema de coordenadas local (el usado en IFC) a uno georreferenciado en CityGML. De esta forma es posible incorporar modelos de mayor detalle, en LOD3 o

³⁶ <https://energyplus.net/>

LOD4, en CityGML, no muy habituales en modelos CityGML, ya que la mayoría suelen ser LOD1 o LOD2.

En el proyecto FASUDIR se desarrolló una herramienta que permitiese simular una ciudad tal cual estaba en la realidad (*baseline*) y diferentes alternativas realizando ciertos cambios en los diferentes elementos de la ciudad. Una alternativa puede ser derruir un edificio y convertirlo en zonas verdes, cambiar el uso de un edificio de oficinas a residencial, aplicar un revestimiento de fachada al edificio, cambiar una carretera por zonas verdes, etc. Con lo cual, era necesario poder representar variantes de una ciudad para después poder lanzar las simulaciones para comparar los resultados. Para ello se han utilizado los servicios que permiten cambiar el tipo de elemento urbano, creación / eliminación de nuevo elemento urbano y cambio de las características de los objetos de la ciudad. Estos servicios permiten cambiar las propiedades de los elementos de la ciudad de una manera ágil y sin tener que generar un nuevo modelo CityGML por cada variante.

Ver la magnitud y complejidad de las operaciones de mantenimiento ha motivado la búsqueda de alternativas más potentes para realizar el mantenimiento, que se describen en más detalle en el capítulo 4.

3.6 Discusión

El flujo de trabajo para la generación de una ciudad 3D descrito se ha subdividido en tres actividades: la configuración del modelo de datos, la generación de la geometría y la generación de la semántica. En otros enfoques, el modelo de datos se configura (creación de una base de datos, identificación de datos temáticos y desarrollo de ADE) después de generar la geometría y la semántica [76].

Se considera que el ciclo de vida de los modelos de ciudad 3D gira alrededor de tres tareas principales: generación, uso y mantenimiento. Hay enfoques que incluyen el almacenamiento como otra tarea adicional [59] [77], mientras que otros ni siquiera contemplan el mantenimiento [78]. En este trabajo se ha englobado el almacenamiento en la tarea de generación y se ha considerado el mantenimiento para así poder disponer de unos modelos debidamente actualizados.

Las actividades de generación de la geometría y la semántica normalmente se realizan por separado, obteniendo de forma directa unos modelos de datos semánticamente enriquecidos. En versiones anteriores del flujo de trabajo de generación, se solía generar

la información geométrica y semántica juntas [77]. Sin embargo, como en algunos proyectos no se disponía de todos los datos semánticos obligatorios, era necesario alterar el código fuente del servicio de generación para cada proyecto. A fin de evitarlo, se ha decidido dividir la actividad en 2 independientes.

La actividad de configuración del modelo de datos incluye la búsqueda de datos geométricos y temáticos. Se ha podido comprobar que esta es una actividad que consume mucho tiempo, ya que implica la búsqueda y el procesamiento inicial de información de diversas fuentes, formatos, grados de precisión y fechas, así como su combinación.

Dada su naturaleza genérica, los parámetros semánticos predefinidos en el núcleo de CityGML son muy básicos, aunque brinda la posibilidad de desarrollar extensiones de dominio de aplicación. Con lo cual se ha generado una ADE que permite incluir en el modelo la información necesaria. No obstante, hay que tener en cuenta que esto requiere de un esfuerzo adicional: antes de poder guardar los datos definidos en las ADE hay que diseñar extensiones de esquema de base de datos, con nuevas tablas, restricciones e índices. Las herramientas de importación/exportación de archivos CityGML a bases de datos espaciales también exigen ciertas modificaciones para poder adaptarlas a las nuevas extensiones [50].

Para terminar, cabe destacar que el mantenimiento a largo plazo de los modelos de ciudad 3D sigue representando un desafío. Para poder comprobar si la propuesta presentada hace la tarea de mantenimiento más factible a un precio razonable no solo deberemos llevar a cabo pruebas de larga duración, sino introducir algunas mejoras para poder hacer frente a los problemas que se vayan destapando. Por ello, en el capítulo 4 se propone un mecanismo para facilitar el mantenimiento de los modelos incluso en escenarios complejos.

4 Mantenimiento colaborativo y despliegue automático de modelos CityGML validados

4.1 Introducción

La creación de modelos CityGML de calidad tiene un coste elevado, tanto en términos de tiempo como de dinero [55]. Las geometrías se pueden generar con distintas tecnologías de adquisición de datos, p. ej. escaneo láser 3D, modelado basado en imágenes o diseño asistido por ordenador [78]. Aunque las geometrías se pueden generar de manera semiautomática, a menudo se requiere cierta cantidad de trabajo manual [60][79]. La principal limitación de los modelos generados con estos métodos es la carencia de información semántica. Algunos autores sostienen que el mayor obstáculo que impide acelerar la adopción de CityGML como un estándar abierto es la dificultad que reviste complementar los modelos 3D con datos semánticos; ya que para ello se necesita cierto grado de intervención manual [80]. La generación de modelos CityGML se detalla en la sección 3.2.4 de esta tesis.

Otro desafío importante alrededor de los modelos de ciudad 3D es el de simplificar su mantenimiento. Los modelos deberían poder actualizarse y ampliarse a medida que las ciudades reales cambian y se dispone de datos actualizados. Si la introducción de modificaciones en los modelos fuese más sencilla, la creación de escenarios del tipo de "*what-if*" y el análisis de alternativas, como las consecuencias que supondrían determinados cambios en un vecindario concreto, sería una práctica más habitual y viable [81].

Incluso en los casos en que los modelos CityGML se someten a un mantenimiento activo, sigue habiendo un gran problema: la mayor parte de los modelos existentes contienen errores geométricos y semánticos [82]. Ponemos a modo de ejemplo un modelo CityGML de libre acceso correspondiente a la ciudad de Rotterdam en el que hay errores de orientación, de intersección de superficies, ausencia de superficies o superficies no planas en el 90% de los objetos de la ciudad [83].

Muchos de los modelos CityGML existentes son generados y mantenidos por los ayuntamientos, ya sea a través de la oficina técnica o de una empresa externa, y están disponibles para su descarga en los sitios web de las localidades. A menudo es una única

persona la que se encarga de este cometido, lo que significa que las tareas de publicación de los modelos actualizados son ejecutadas y controladas por un solo participante.

La creación y mantenimiento colaborativos de los modelos presenta varias ventajas, entre ellas unos resultados más inmediatos gracias al trabajo en paralelo y reparto de los esfuerzos, o la posibilidad de que expertos en diferentes materias cooperen y aporten sus conocimientos en distintos aspectos del mismo modelo. Sin embargo, la creación y el mantenimiento colaborativo supondría problemas adicionales, como la integración de modificaciones en conflicto.

En este capítulo se presenta una solución para facilitar el mantenimiento regular de unos modelos de ciudad 3D en CityGML. La solución se fundamenta en la estrategia y las herramientas de despliegue continuo empleadas en el desarrollo de *software* y se adapta al problema derivado de crear, mantener y desplegar modelos CityGML validados. Además, en esta sección también se demuestra cómo esta solución ayudaría a superar los desafíos y problemas descritos anteriormente: 1) reduce la carga de trabajo manual gracias a la automatización de los procesos, 2) fomenta el mantenimiento colaborativo de los modelos por medio de la integración de un sistema de control de versiones (VCS, *Version Control System*) en el flujo de trabajo y 3) reduce las probabilidades de errores geométricos o semánticos con ayuda de comprobaciones automáticas frecuentes.

4.2 Trabajo relacionado

Ahora mismo el proyecto Open Street Map (OSM) es un ejemplo excelente de creación y mantenimiento colaborativos en relación a modelos geográficos en 2D georreferenciados [84]. OSM dispone de sus propias herramientas y modelo de datos, con soporte *ad hoc* integrado para la edición multiusuario y el control de versiones. En el caso de las ciudades 3D, se han lanzado varias propuestas. Una de ellas es una metodología para el desarrollo colaborativo e interactivo de modelos de edificios 3D que contempla muros, tejados, puertas y ventanas como elementos estructurales [85]. Otros autores han optado por mejorar los datos de OSM para compatibilizarlos con la visualización tridimensional. Por ejemplo, la plataforma OSM-3D ha sido desarrollada precisamente para ofrecer una vista 3D interactiva de los datos *online* [86] [87]. No obstante, hay que tener en cuenta que la tridimensionalidad de OSM se basa en la extrusión, de modo que no sirve para representar geometrías complejas. Otra solución es 3D Repo, un entorno de trabajo de código abierto pionero para el control de versiones que permite la gestión de

manera coordinada de datos 3D de ingeniería a gran escala a través de Internet, aunque por el momento no es compatible con CityGML [88].

Los modelos de datos que captan ciertos aspectos de la realidad no son útiles para ciertas aplicaciones si no se actualizan con regularidad. Se han llevado a cabo estudios sobre los factores de obsolescencia en el caso de los modelos de ciudades 3D, a raíz de los cuales se han propuesto algunas estrategias preventivas. Entre ellas se incluye la toma en consideración de factores como el formato de los archivos, la interoperabilidad de los datos, la accesibilidad o la usabilidad.

Además, tampoco hay que olvidar que el mantenimiento regular de los modelos de ciudades generaría un historial de cambios que pueden ser de gran utilidad a la hora de analizar la evolución de una ciudad a lo largo del tiempo. Pese a todo, el estándar CityGML actualmente no ofrece ninguna solución para almacenar este tipo de información, ya que solo incluye los datos relativos al estado actual [66]. Una alternativa es mejorar CityGML para permitir que las versiones del modelo en su conjunto, o de elementos específicos del mismo, puedan representar diferentes alternativas de planificación [63]. Existe un nuevo enfoque de modelado y la voluntad de implementar soporte que permita la gestión de versiones e historiales en CityGML [63]. Otro problema estrechamente relacionado con esto es mantener las modificaciones bajo control a medida que los modelos van evolucionando. Las modificaciones están completamente bajo control cuando se sabe quién ha cambiado qué, cuándo y por qué motivo. Además del conflicto de las versiones, este problema ya se identificó en los inicios de CityGML [65], pero aún no se ha abordado en el estándar.

En lo que al grado de corrección de los modelos CityGML respecta, existen numerosas propuestas para simplificar la detección de errores y la depuración de los mismos. La detección de errores geométricos en modelos de ciudad 3D es imprescindible de cara a poder garantizar operaciones de procesamiento o manipulación, tales como el cálculo de los volúmenes construidos, los muros adyacentes o la radiación solar [89][90]. El grupo de trabajo *SIG 3D Quality* ha definido una serie de directrices con recomendaciones para el correcto modelado, para así evitar errores en los modelos CityGML [91]. Sin embargo, se necesitan ciertas reglas para validar la consistencia geométrico-semántica del modelo, incluso cuando los datos cumplen el estándar [92] [93]. Alam *et al.* han desarrollado una herramienta para reparar los errores detectados en los modelos CityGML de modo

automático [94]. Val3dity es otra herramienta diseñada para validar formas 3D primitivas en base al estándar internacional ISO19107 [95]. CityDoctor, por su parte, implementa métodos y cálculos para el análisis, comprobación y corrección de la sintaxis, geometría y semántica de modelos de ciudad 3D virtuales [96]. Actualmente ninguna de estas herramientas puede detectar y eliminar todos los posibles errores dentro de un modelo CityGML. El experimento de interoperabilidad de calidad CityGML (*CityGML Quality Interoperability Experiment*) presenta una serie de recomendaciones sobre requisitos de conformidad relativos a herramientas de comprobación de la calidad, así como un flujo de validación [97].

4.3 Problemática de la creación y mantenimiento colaborativo de modelos urbanos

En el escenario hipotético que se describe a continuación (ver Figura 4-1) se presentan las problemáticas habituales a la hora de crear y mantener de forma colaborativa modelos urbanos entre diferentes agentes. En el escenario se presentan diferentes agentes que tienen diferentes roles (ayuntamiento, empresas, ciudadanos) y se describen acciones hipotéticas entre los mismos de cara a poder identificar las principales problemáticas. En el siguiente diagrama el tiempo transcurre de arriba abajo. Además, los nombres de los diferentes actores se representan con la primera letra (p.e. Arturo es A, Bisores S.A. es B, etc.) y las diferentes versiones y variantes también se identifican (p.e. II, III, IIIa, etc.):

1. Arturo trabaja en el ayuntamiento y mantiene un modelo CityGML de la ciudad (I). Hace los cambios con el editor CityEditor³⁷, lo prueba cargándolo en un visor y, de vez en cuando, lo sube a la web para que lo puedan descargar los vecinos.
2. Llega una contrata Bisores S.A. y hace un visor web del modelo CityGML. Ahora además de subirlo a la web, Arturo tiene que probar que el modelo (II) funciona bien en el visor; ya que el visor solo visualiza en LOD2, pero el modelo del ayuntamiento está en LoD3, ya que lo exige el ayuntamiento.
3. El ayuntamiento contrata a Carmen para diseñar como podría ser el nuevo parque que se va a construir donde la antigua cárcel. Arturo le pasa el modelo CityGML (II) para que Carmen lo pueda modificar y generar distintas versiones para mostrarlas al público y al departamento de urbanismo.

³⁷ <https://extensions.sketchup.com/en/content/cityeditor-2>

4. Mientras Carmen trabaja en lo suyo, corrige algunos errores que ha detectado en varios edificios del entorno de la antigua cárcel. Alguno de estos errores es grave, y le pasa una copia del CityGML (III) a Arturo para que lo integre con lo suyo.
5. Arturo está haciendo otros cambios, y no se atreve a integrar los de Carmen hasta terminar, así que los errores tardan un tiempo en poder ser corregidos. Además, dado que comprobar que todo va bien es un proceso manual, el CityGML corregido (III) aún tarda un tiempo en poder estar publicado y accesible en el visor. Arturo para comprobar que todo está bien tiene que abrir el modelo con su herramienta, comprobar que la geometría del modelo es correcta, ver qué la información semántica está completada y si está bien completada, etc.
6. El ayuntamiento hace un concurso de ideas para la reforma del entorno del Convento de los Pesares. Para ello se partirá del CityGML publicado (III), se generará una propuesta y se enviará al ayuntamiento por e-mail (se lo hará llegar) antes de un plazo límite. Las consultoras DatosGeo S.L., Estudios integrales S.L. y Furbicom S.A. descargan el modelo y empiezan a trabajar.
7. Carmen ha generado 3 propuestas distintas (IIa, IIb y IIc) y es necesario hacerlas públicas. Arturo integra a mano las 3 propuestas con el modelo "principal", las comprueba a mano y las sube a la web (III+IIa, III+IIb y III+IIc). Además le pide a Bisores S.A. que modifiquen la app web para poder mostrar varios modelos seleccionando en un menú.
8. DatosGeo termina su propuesta (IIIId) y la envía por e-mail. Arturo la recibe pero como está a mitad de un cambio importante en la parte norte de la ciudad, la deja pospuesta.
9. La consultora Furbicom, para elaborar el proyecto organiza un equipo de trabajo multidisciplinar. Cada uno de los miembros del equipo utiliza un submodelo de la versión descargada que puede estar acotado en extensión o temática. Una vez recibidos las propuestas partes de los equipos de trabajo, la consultora es la encargada de testear e integrar cada una de las partes. Esta integración puede provocar inconsistencias que requieren actualizar las propuestas parciales. Estudios integrales y Furbicom terminan sus propuestas (IIIe y IIIf) y las envían por e-mail. Arturo las deja también en espera a la espera de poder integrarlas.
10. Tras 10 días, Arturo ha terminado de modificar la parte norte e integra la propuesta de DatosGeo. Lo primero que hace es chequear que sea correcta, pero detecta que hay varios errores importantes y les avisa que los tienen que corregir. Detecta que

la geometría está representada de manera distinta a la utilizada por el ayuntamiento. Las herramientas del ayuntamiento necesita la geometría representada con sólidos, sin embargo el modelo de DatosGeo no lo representa de esa forma. DatosGeo protesta porque han pasado varios días y se les podía haber avisado antes, y que además no estaba nada claro en el concurso qué tenía que tener el modelo para ser correcto y que en su software el modelo se ve bien.

11. La asociación de vecinos del barrio de la antigua cárcel ha visto una de las propuestas de Carmen y, como su presidente Gonzalo es un manitas, carga el modelo (III+IIa) en su herramienta favorita y corrige algunas pequeñas omisiones que ha detectado (son cosas recientes del entorno de la cárcel que el modelo actual de la ciudad no tenía contempladas). Añade también un pequeño puente peatonal que se ha construido recientemente en el entorno de la cárcel. Manda el fichero (III+IIa+I) por e-mail a Carmen sugiriendo que podrían incorporarlas.
12. Carmen revisa la propuesta de Gonzalo y sustituye la que había hecho ella por la que ha recibido. Se la pasa a Arturo por correo (III+IIa+I) para que la suba a la web. Al mismo tiempo Arturo revisa las propuestas de las otras dos consultoras y están bien.
13. Arturo coge la propuesta de Carmen actualizada con lo de Gonzalo y lo testea antes de subirlo a la web. En ese momento se da cuenta de que la versión de Gonzalo está en una versión superior (la v2.0.0, que incluye un puente) a la que se están utilizando (v1.0.0). Ni el ayuntamiento ni el visor soportan esta versión, con lo cual Arturo pide a Carmen que vuelva a la versión anterior sin las incorporaciones de Gonzalo. Gonzalo muestra su disconformidad. Arturo se da cuenta de la limitación y pide al ayuntamiento portar el sistema al v2.0.0.
14. Definido el ganador de la oferta, se integra la propuesta de la consultora F con el “principal” y se sube a la web (III+III_f). El resto de propuestas se desechan.

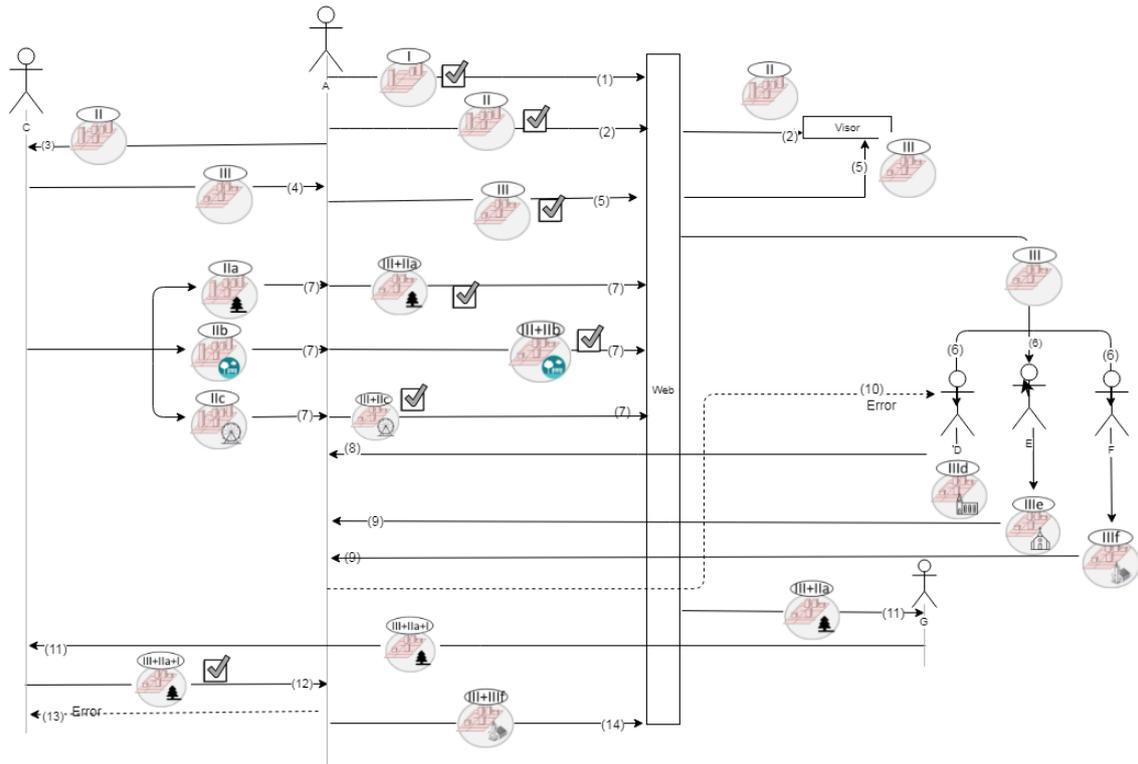


Figura 4-1 Escenario hipotético

Los principales problemas identificados en el escenario hipotético son los siguientes (ver Tabla 4-1):

Tabla 4-1 Problemas identificados en el escenario hipotético

Id	Nombre	Descripción	Punto en el escenario
P1	Despliegues manuales	Cada vez que el ayuntamiento quiere desplegar o hacer público un modelo CityGML, tiene que realizarse de forma manual	1, 2, 5, 7, 14
P2	Integración manual	La integración de los cambios en el CityGML principal se realiza de forma manual. En caso de conflictos es un proceso muy complicado de realizar.	4, 5, 7, 10, 12, 14
P3	Errores que surgen en distintos momentos y solo se descubren con pruebas manuales	Los errores geométricos, semánticos, de requisitos, de visualización, etc. se descubren al realizar pruebas manuales. Sin embargo, muchos de estos errores podrían identificarse de forma sencilla.	4, 5, 10, 13
P4	Criterios de corrección/calidad poco claros	Cada agente o empresa tiene unos criterios de corrección y de calidad. Con lo cual, a la hora de colaborar con diferentes agentes, existen mal entendidos en este aspecto que implican rehacer trabajo.	10, 13

P5	Mantenimiento de alternativas totalmente manual	de	En el caso de trabajar en forma paralela en alternativas, el mantenimiento de las mismas se hace de forma manual. Y los cambios en cada una de las alternativas no es controlable.	7
P6	Difícil seguir cambios en los datos y el modelo		Al no haber un registro de los cambios, es complicado saber quién ha cambiado, que, cuando y por qué.	5, 7, 10, 12, 14
P7	Trabajo precario	remoto	El intercambio de ficheros se hace utilizando e-mails, pero tanto podrían ser Dropbox o similares, lo que conlleva un trabajo remoto precario.	6, 8, 9, 11, 12
P8	El exceso de trabajo manual impone retrasos en distintos momentos y por distintas razones		Como todas las validaciones pasan por un único agente, forma un cuello de botella y la consecuencia es que se imponen retrasos constantemente.	5, 8, 9, 10, 12

Utilizando sistemas de control de versiones se solucionan los problemas de trabajo remoto precario, es más fácil seguir los cambios realizados y el mantenimiento de alternativas está contemplado, con lo cual los problemas P5, P6 y P7 se solucionan. Mediante validaciones automáticas es posible detectar los errores, minimizar las pruebas manuales, los criterios de calidad y corrección serán las mismas en todo momento y validar el modelo no creará retrasos, con lo cual los problemas P3, P4 y P8 se solucionan. Finalmente, controlando y automatizando de forma sistemática la validación y el despliegue, la integración ya no es manual y los despliegues tampoco, con lo cual los problemas P1 y P2 se solucionan.

4.4 Despliegue continuo de modelos CityGML

En este apartado se describe un método para automatizar el proceso de validación y despliegue de modelos CityGML y fomentar el mantenimiento colaborativo de los modelos por parte de varias personas. La aplicación de este método, que se fundamenta en la estrategia de despliegue continuo empleada para el desarrollo de *software*, a modelos CityGML, consiste en 2 tareas principales:

- (1) Diseñar e implementar un *pipeline* de despliegue para el modelo en cuestión. En resumen, se trata de una secuencia de pasos automáticos vinculados entre sí dirigidos a transformar los datos de entrada, es decir, un modelo CityGML modificado en un repositorio del VCS, en un modelo CityGML validado y desplegado. Una opción sería implementar estos *pipelines ad hoc*, p. ej. desarrollando funciones de validación en cualquier lenguaje de programación, sin embargo, de esta forma se limita la reusabilidad e implica grandes cambios cada vez que se cambian los requisitos. Es mejor aprovechar herramientas especializadas, ya que de esta forma solamente es necesario desarrollar los servicios de validación una única vez por cada modelo de ciudad y después es posible reutilizar ciertas partes del *pipeline*.
- (2) Ejecutar el *pipeline* en cuestión cuando se ha realizado un *commit* en el VCS ya que se han realizado una serie de modificaciones en un modelo CityGML. Si bien la frecuencia dependerá de la cantidad y complejidad de las modificaciones y del número de personas que intervengan simultáneamente en el modelo, si se toma como referencia los procesos de integración continua de software, se podría considerar normal compartir las modificaciones y ejecutar el pipeline sobre todos los cambios realizados al modelo al menos una vez al día.

En el resto de este apartado se explica cómo diseñar e implementar estos *pipelines* de despliegue. Dado que el método propuesto se basa en el despliegue continuo, se comienza por describir esta estrategia de desarrollo de *software*.

4.4.1 Despliegue continuo

El despliegue continuo (CD, *Continuous Deployment*) es una metodología empleada para controlar y automatizar de forma sistemática la comprobación y el despliegue o entrega de *software* y servicios [98]. Aunque el CD se emplea sobre todo para el desarrollo de

software, existen algunos casos en los que se ha aplicado para el despliegue de otros elementos, p. ej. documentos [99] [100].

El CD está basado en el concepto de los *pipelines* de despliegue, que constituyen la implementación automática del proceso de desarrollo, comprobación y despliegue de un sistema de *software*. Los *pipelines* de despliegue constan generalmente de varias fases (aunque también pueden tener solo una): cada una de ellas es más lenta que las anteriores, pero garantiza de forma fiable la exactitud del *software* que se está desplegando. Esta creciente fiabilidad viene dada por la ejecución automática de unas comprobaciones cada vez más complejas. Para implementar un *pipeline* de despliegue se necesita al menos un repositorio de control de versiones, en el que los desarrolladores deben guardar las modificaciones que introduzcan en el código.

Un *pipeline* de despliegue básico comienza por la fase de *commit* de modificaciones, en la que se guardan ciertos cambios en un repositorio de códigos fuente, se compilan y ensamblan los códigos binarios (p. ej. en ejecutables) y se llevan a cabo algunas comprobaciones básicas. Si surgen fallos se notifica al programador y si, por el contrario, se desarrolla satisfactoriamente, se procede a la fase de *acceptance*: donde se configura el entorno de comprobación en el que se despliegan los códigos binarios y se realizan unas pruebas de aceptación. Una vez superada esta fase, se pasa a la fase de *production* final, en la que se configura el servidor de producción y se despliegan los códigos binarios para que el usuario final pueda ejecutarlos.

Se pueden añadir otras fases, p. ej. una fase de *capacity* para medir el rendimiento del *software* desplegado o una fase de *user acceptance test* que consistente en pruebas, incluso manuales, orientadas al usuario.

4.4.2 Pipeline de despliegue de CityGML

La propuesta que se presenta en este capítulo implica el diseño y la implementación de *pipelines* de despliegue para los modelos CityGML que se vayan a crear, mantener y desplegar. Estos *pipelines* deben seguir la misma estructura que los *pipelines* de despliegue de los sistemas de *software* descritos en el apartado anterior, aunque muchas de las tareas difieren, ya que los problemas a resolver son distintos. Al igual que ocurre con los *pipelines* de despliegue de *software*, los *pipelines* CityGML deben diseñarse a la

medida de cada modelo de ciudad, aunque comparten una estructura común con un máximo de 4 fases, como se ilustra en la Figura 4-2.

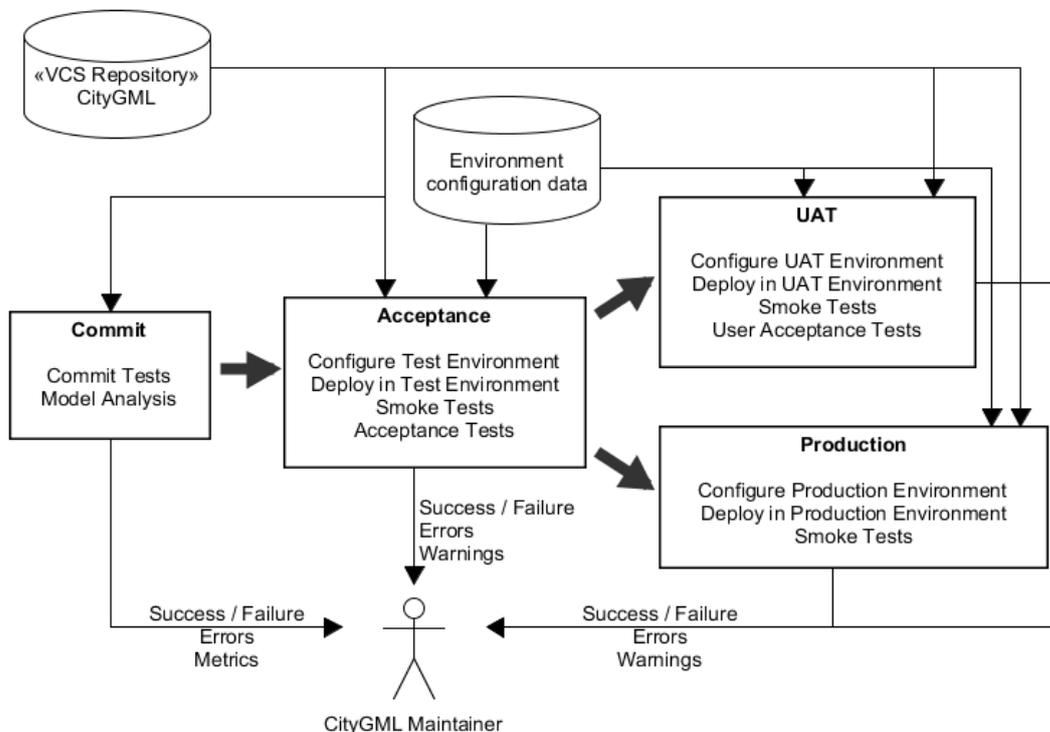


Figura 4-2 Pipeline de despliegue CityGML

- **Fase de *Commit*:** Sirve para comprobar si un archivo CityGML está correctamente estructurado y si cumple el estándar. Esta fase toma como datos de entrada los archivos CityGML confirmados en el repositorio del VCS y los examina en busca de errores estructurales y no conformidades respecto al estándar CityGML. Además, se realizan también ciertos cálculos estadísticos de los elementos contenidos en los archivos. Si el archivo CityGML es correcto, esta fase vuelca una serie de cálculos y estadísticas al archivo en cuestión. Sin embargo, si es incorrecto, el resultado es un listado de errores.
- **Fase de *Acceptance*:** La finalidad de esta fase es desplegar el modelo de ciudad en el mismo formato que se necesita para el despliegue orientado al usuario final en el marco de un entorno de prueba. En este entorno se ejecutan varias comprobaciones de aceptación para garantizar que el modelo cumple los requisitos del usuario en un entorno similar al entorno final de producción. En esta

fase los datos de entrada son archivos CityGML estructuralmente correctos. Las pruebas que se llevan a cabo se basan en los criterios de aceptación, que están, a su vez, basados en los requisitos del usuario. Esta fase genera un listado de errores y advertencias relativas a las comprobaciones realizadas. Cuantos más criterios de aceptación se definan e implementen a modo de pruebas automáticas en esta fase, mayor confianza se puede tener en que el modelo CityGML cumpla los requisitos del usuario en el entorno de producción.

- **Fase de *User Acceptance Test* (UAT):** Cabe la posibilidad de que algunas de las pruebas de aceptación se deban realizar manualmente debido a que aún no se hayan automatizado o que no se puedan automatizar (p. ej. la exploración visual de un modelo para detectar inconsistencias). Es en esta fase donde se asegura que esas pruebas manuales de aceptación del usuario se llevan a cabo. En ella, se despliega un archivo CityGML (previamente comprobado durante las fases de confirmación y aceptación) en un entorno de prueba para la aceptación del usuario. El evaluador es informado de que hay pruebas manuales pendientes de ejecutar y el *pipeline* se queda en pausa hasta que se hayan completado y marcado como "aptas" o "no aptas". Después, la fase se da por terminada. Esta fase emite un informe en el que se recogen los resultados obtenidos y, en caso de fracaso, un listado con los errores detectados y comentarios del evaluador.
- **Fase de *Production*:** esta fase sirve para desplegar el modelo de ciudad en el formato solicitado por el usuario final en el marco de un entorno de producción. En ella se toma como datos de partida un archivo CityGML que haya superado las fases de confirmación, aceptación y UAT, se configura un entorno de producción en el que después se despliega el modelo de ciudad y se ejecutan algunas pruebas de humo para verificar que el despliegue no ha sido fallido. Al final se genera un informe con los resultados del proceso y, en caso de ser satisfactorio, el archivo CityGML queda a disposición del usuario final. En este capítulo, el término "despliegue" se emplea en su sentido más amplio. Puede hacer referencia a colocar el *software* en un directorio accesible desde un servidor web, a integrarlo en una aplicación web que lo ejecute, posiblemente a través de la interfaz de un servicio web, o incluso a empaquetarlo como un recurso dentro de una aplicación móvil antes de publicarlo en el mercado de aplicaciones.

En los siguientes apartados se describen las tareas que se llevan a cabo en estas fases en más detalle.

4.4.2.1 *Fase de Commit*

Commit tests

La validación de la correcta estructura del archivo CityGML implica: (1) la validación del esquema XML (versión XML, XSD y CityGML), (2) la validación geométrica para asegurarse de que la representación de la información geométrica cumple el estándar GML, y (3) la validación semántica, incluida la revisión de los listados de código para atributos enumerativos.

Model analysis

Con esta tarea se calculan varias estadísticas descriptivas en base al archivo CityGML. Entre estos cálculos se incluye el número de elementos contenidos en el modelo, los elementos de cada nivel de detalle (LOD) y el número de elementos que incluyen un conjunto básico de atributos (p. ej. clase, función y uso).

4.4.2.2 *Fase de Acceptance*

Configure test environment

La primera tarea de esta fase es crear el entorno en el que se llevan a cabo las pruebas de aceptación. Los datos para la configuración del entorno deberían incluir los ajustes necesarios para componentes del *software* como el servidor de la base de datos y el servidor web.

Deploy in test environment

Esta tarea despliega el modelo de ciudad en el formato requerido para despliegue para el usuario final. Por ejemplo, una base de datos o un sistema de archivos serían algunas opciones que se podría emplear para el despliegue.

Smoke tests

Las pruebas de humo son comprobaciones sencillas y rápidas que sirven para verificar que un proceso ha sido ejecutado sin fallos graves. En este caso, van dirigidas a comprobar que el proceso de despliegue en el entorno de prueba ha concluido sin errores

críticos. Si se toma la base de datos como ejemplo, bastaría con lanzar un sencillo comando de recuperación de datos. Si el despliegue se realiza sobre un archivo guardado, la prueba de humo podría verificar si dicho archivo se ha subido correctamente al repositorio (p. ej. comprobando si el tamaño del archivo cargado es correcto).

Acceptance tests

Esta tarea sirve para comprobar si el modelo de ciudad cumple las especificaciones de aplicación y del cliente, expresadas en forma de criterios de aceptación basados en dichos requisitos, es decir basados en problemas concretos, en el marco de un entorno similar al de producción. Por ejemplo, esta tarea generalmente verifica que el modelo incluye todas las propiedades semánticas exigidas (área y orientación principal de las fachadas y muros adyacentes, etc.). Además, estas pruebas también permiten revisar la jerarquía de edificios. Pongamos por ejemplo que en cierto sector se exige que los edificios dispongan de información sobre la parcela y geometría en LOD0 y LOD1 y que las partes de los edificios contengan información de construcción real y geometría en LOD2. Otras pruebas podrían consistir en verificar que la altura de la geometría y la altura especificada en el parámetro de altura medida "*measuredHeight*" coinciden, o que los muros estén representados con superficies verticales y el terreno y el tejado con superficies más horizontales.

4.4.2.3 Fase de UAT

Configure, Deploy and Run Smoke Tests in the UAT Environment

Las acciones que esta fase engloba son similares a las descritas en la fase de aceptación, a excepción de que, en este caso, se ejecutan dentro del entorno UAT.

User Acceptance Tests

Estas pruebas permiten verificar si el modelo de ciudad desplegado es realmente válido desde la perspectiva del usuario, considerando elementos que resultan muy difíciles o incluso imposibles de expresar en forma de criterios de aceptación verificables de manera automática, por lo que se deben ejecutar manualmente. En primer lugar, el modelo CityGML se puede examinar como un elemento aislado. Aunque la información geométrica sea correcta según el estándar, cabe la posibilidad de que esta no sea válida (p. ej. porque no se representan los patios existentes o algunas partes del modelo carecen

de textura o presentan un mapeado de textura inadecuado). Después de examinar el modelo como un elemento aislado, se puede proceder a comprobar su integración en el entorno final, tomando en consideración, entre otros, su geolocalización y su orientación. El evaluador es informado de que hay pruebas manuales pendientes de ejecutar y el *pipeline* queda en pausa hasta que se hayan completado y marcado como "aptas" o "no aptas".

4.4.2.4 Fase de Production

Configure, Deploy and Run Smoke Tests in the Production Environment

Las acciones en esta fase son similares a las descritas en la de aceptación, solo que, en este caso, se ejecutan dentro del entorno de producción.

4.4.3 Implementación de pipelines de despliegue de CityGML

En este apartado se aborda la implementación de *pipeline* de despliegue CityGML como los descritos en el apartado anterior. La principal estructura de estos *pipelines* se puede complementar con un sistema de control de versiones y cualquier herramienta de despliegue continuo. Estas herramientas están muy extendidas y bien documentadas hay una gran variedad entre las que elegir. De hecho, una de las ventajas de haber basado esta propuesta en los *pipelines* de despliegue de *software* es la posibilidad de sacar el máximo provecho de ellas. Las otras herramientas que se necesitan son aquellas que ofrezcan la funcionalidad correspondiente a las diferentes tareas de las distintas fases que constituyen los *pipelines* de despliegue, teniendo en cuenta que la mayoría de ellas están vinculadas específicamente a CityGML. Estas herramientas son el equivalente aproximado a los compiladores, enlazadores, analizadores de código y entornos de prueba que se emplearía en un *pipeline* de despliegue de *software*. Algunas herramientas de estas han sido desarrollos propios en CityGML, mientras que otras son adaptaciones de otras herramientas ya existentes.

En lo que a la estructura de los *pipelines* respecta, lo primero que se necesita es un VCS. Los sistemas de control de versiones sirven para registrar los cambios que se van realizando en los archivos con el tiempo, para poder recuperar versiones específicas más adelante. Actualmente, sistemas de control de versiones distribuidos (DVCS, *Distributed Version Control Systems*) como Git o Mercurial están ganando cada vez más popularidad y ofrecen un sinfín de novedosas capacidades. Los DVCS permiten a los desarrolladores

trabajar en copias locales de los repositorios, es decir, fuera de línea, al tiempo que conservan todo el historial del proyecto al completo, además de crear y fusionar ramas de forma más económica y compartir solamente los conjuntos de modificaciones que ellos seleccionen.

Además del VCS, también se necesita una herramienta de CD para implementar la estructura del *pipeline*. Existen muchas entre las que elegir, aunque algunas de las más habituales son Jenkins, Team City, Cruise Control, Bamboo y GoCD.

La Tabla 4-2 recoge varias herramientas desarrolladas exclusivamente para la implementación del *pipeline*, con una breve descripción y sus datos de entrada y salida (algunas son específicas para CityGML). Estas herramientas han sido empaquetadas en Java como sencillos servicios web REST fácilmente ejecutables desde una herramienta de CD y no son específicos de un modelo o *pipeline* CityGML concreto, de modo que se pueden reutilizar. El conjunto de herramientas implementado actualmente pretende ofrecer una serie de funcionalidades que permitan implementar un *pipeline* de despliegue CityGML representativo a fin de validar la propuesta presentada.

Tabla 4-2 Herramientas desarrolladas exclusivamente para la implementación del *pipeline*

Fase de *commit*

Nombre	Descripción	Entrada	Salida
Comprobador de versiones CityGML	Comprueba los <i>namespace</i> del CityGML para determinar si hacen referencia a la versión 2.0.0 de CityGML.	Archivo CityGML	Versión válida o listado de errores
Comprobador de geometría CityGML	Actualmente comprueba que el primer y último puntos de cada superficie son iguales y que no hay puntos repetidos consecutivamente.	Archivo CityGML	Geometría válida o listado de errores
Comprobador de semántica CityGML	Valida el modelo contra los listados de código CityGML definidos.	Archivo CityGML	Listados de código CityGML correcto o listado de errores
Generador de estadísticas CityGML.	Calcula cuántos de los edificios incluidos en el modelo CityGML contienen cada uno de los siguientes elementos cumplimentados: clase (<i>class</i>), función (<i>function</i>), año de construcción (<i>year of construction</i>), tipo de tejado (<i>roofType</i>), altura medida (<i>measuredHeight</i>), número de pisos hacia arriba (<i>storeysAbove</i>) y número de pisos hacia abajo (<i>storeysBelow</i>).	Archivo CityGML	Cálculo o listado de errores

Fase de *acceptance*

Nombre	Descripción	Entrada	Salida
Comprobador de la semántica de las fachadas	Comprueba que se han calculado los parámetros de área y orientación principal de todas las fachadas.	Archivo CityGML	Parámetros calculados o listado de errores

Fases de *acceptance*, *UAT* y *production*

Nombre	Descripción	Entrada	Salida
Generador de solicitudes WFS	Envía una solicitud <i>GetCapacities</i> para recuperar las capacidades y comprueba que el servicio WFS se ha configurado correctamente a modo de prueba de humo.	URL del WFS	Configuración del WFS o listado de errores

La Tabla 4-3 recoge las herramientas y librerías de terceros empleadas. Mientras que algunas son específicas para CityGML, otras son más genéricas y se han adaptado como corresponde para poder usarlas en *pipelines* de despliegue CityGML, empaquetándolas como servicios web REST en Java.

Tabla 4-3 Librerías de terceros

Fase de *commit*

Nombre	Descripción	Entrada	Salida
Analizadores sintácticos de Java XML	Analiza la sintáctica del archivo de CityGML para determinar si está bien formado.	Archivo CityGML	Archivo CityGML bien formado o listado de errores
SchemaFactory de Java	Analiza la sintáctica del archivo CityGML para determinar si el CityGML cumple el esquema.	Archivo CityGML	Archivo CityGML con esquema válido o listado de errores

Fase de *UAT*

Nombre	Descripción	Entrada	Salida
Validador de integración de los elementos entorno real	El modelo CityGML se visualiza en una aplicación web para ver los mapas y la Tierra en 3D.	URL del archivo CityGML exportado a KML	Archivo CityGML visualizado en Cesium ³⁸

³⁸ Cesium es una aplicación web para ver mapas y terreno en 3D. <http://cesiumjs.org/>

Fases de *acceptance*, UAT y *production*

Nombre	Descripción	Entrada	Salida
3DCityDB - Generador de estructura de bases de datos	- Crea el esquema SQL necesario para guardar un modelo CityGML.	Archivo de órdenes PostgreSQL + sistema de referencia de coordenadas + detalles de conexión y configuración (p. ej. anfitrión, puerto, usuario, contraseña)	3DCityDB Base de datos generada con estructura CityGML o listado de errores
3DCityDB - Importador de CityGML a DB	- Importa el modelo CityGML a la base de datos.	Archivo CityGML + detalles de conexión y configuración (p. ej. anfitrión, puerto, usuario, contraseña)	Archivo CityGML guardado en la base de datos o listado de errores
Deegree	El WFS se ejecutando configuración de la base de datos, las funciones a poner a disposición y el servicio.	Nombre del proyecto + detalles de conexión y configuración (p. ej. anfitrión, puerto, usuario, contraseña)	WFS configurado o listado de errores

4.5 Escenario de ejemplo

En este apartado se aplica la solución propuesta en este capítulo, basada en *pipelines* de despliegue CityGML, en un escenario que representa el tipo de problemas que pretende resolver. Este escenario sirve tanto para probar la viabilidad de la propuesta, como para ilustrar sus posibilidades. Se fundamenta en datos reales y trabajo que se está llevando a cabo actualmente para la rehabilitación energética de edificios [101].

En este escenario varios participantes colaboran en el mantenimiento de un modelo de ciudad que se emplea, entre otras finalidades, para identificar las mejores soluciones para la rehabilitación energética de un distrito. El escenario está sito en la ciudad de Santiago de Compostela (España), para la que se creó un archivo CityGML. Este modelo CityGML

se somete a frecuentes actualizaciones y es importante que preserve su exactitud en todo momento. El mantenimiento del modelo CityGML recae en las siguientes personas:

- **Arquitecto municipal:** es responsable de la conservación de los elementos principales de CityGML. Se encarga de actualizar los modelos CityGML periódicamente con datos facilitados por otras instituciones (p. ej. el catastro o la agencia nacional de cartografía). El Ayuntamiento solo está familiarizado con unas propiedades semánticas básicas y la geometría de la ciudad a bajo nivel de detalle (LOD0, LOD1 y LOD2).
- **Técnica de sistemas de información geográfica externa al Ayuntamiento:** ella es responsable de completar y mejorar el nivel de detalle del modelo de ciudad añadiendo los elementos que faltan y mejorando el nivel de detalle de los que ya existen hasta LOD3 (que incluye ventanas y puertas).
- **Consultor de energía:** su cometido es completar la información semántica del modelo CityGML con algunas extensiones. Para conseguir dicha información, debe desarrollar trabajo de campo a diario, sobre todo realizar inspecciones visuales de los edificios y editar el CityGML. Esta información puede hacer referencia al edificio en sí mismo o a sus elementos (p. ej. año de construcción, función del edificio, principal material de construcción, instalaciones, porcentaje de acristalamientos y materiales).

4.5.1 Configuración

La Figura 4-3 muestra la configuración del escenario de ejemplo. Existe un repositorio Git compartido alojado en Github, que es donde se comparte la última versión correcta del archivo CityGML. Los participantes tienen permiso de lectura y escritura en el repositorio, es decir, que siguen el denominado "flujo de trabajo centralizado" en Git [102]. Todos cuentan con un clon local del repositorio compartido, en el que extraen el repositorio e introducen sus modificaciones a medida que trabajan. Además, todos ellos disponen de un servidor Jenkins CI en sus ordenadores locales donde pueden ejecutar una versión simplificada del *pipeline* de despliegue compartido, que de momento solo contiene la fase de *commit*, para verificar que esas modificaciones no han frustrado ninguna prueba de asignación antes de pasar a compartirlo con el resto. Una vez que se supera la fase de *commit* del *pipeline* local abreviado con éxito, ya es posible compartirlo a través del repositorio Github.

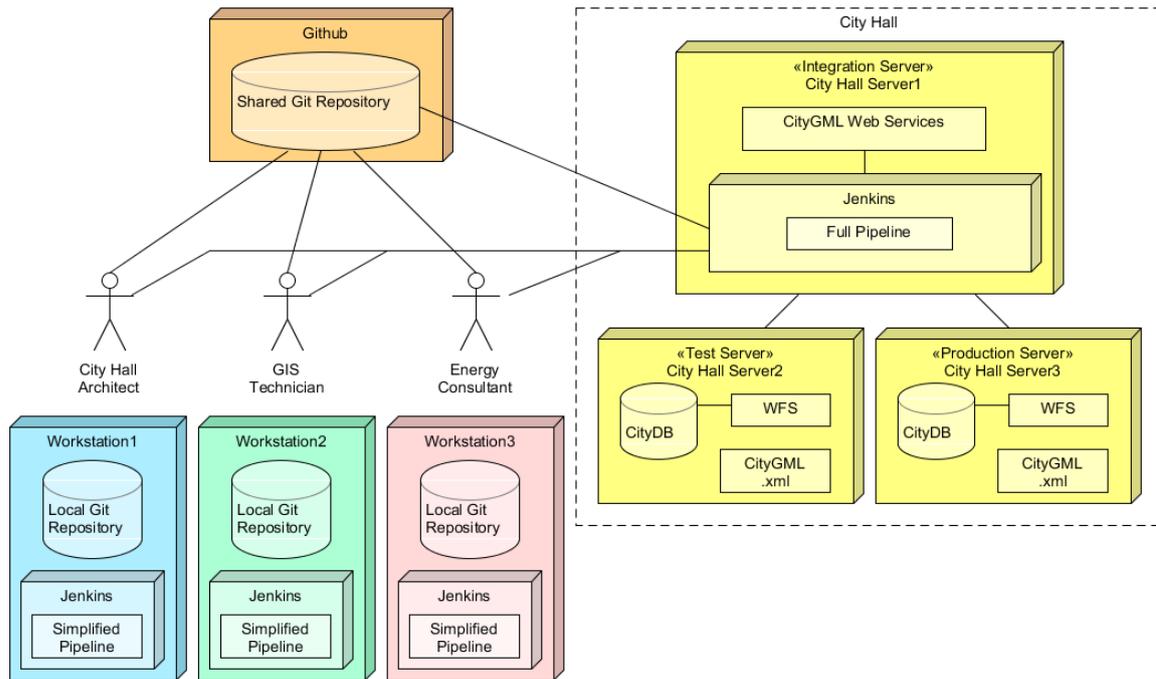


Figura 4-3 Configuración del escenario de ejemplo

El repositorio de Github está vinculado a un *pipeline* de despliegue completo en una instancia de un servidor Jenkins CI alojado en un ordenador dentro del Ayuntamiento (servidor de integración). El servidor de integración, por su parte, tiene acceso a un servidor de prueba, en el que se ejecutan las pruebas de las fases de aceptación y UAT, y a un servidor de producción, donde la nueva versión del archivo CityGML es desplegada y, si logra atravesar el *pipeline* completo con éxito, puesta a disposición de la ciudadanía. Estos servidores, que disponen de una base de datos PostgreSQL con extensión PostGIS y un servicio WFS, sirven para publicar el modelo CityGML. Además, el CityGML también se despliega en forma de archivo descargable para aquellos usuarios que lo necesitan así. El *pipeline* completo se acciona automáticamente con cada envío al repositorio compartido.

La Tabla 4-4 enumera las herramientas empleadas a lo largo de todo el *pipeline* completo (escogidas entre todas las herramientas descritas 4.4.3):

Tabla 4-4 Tareas de cada fase en el marco del pipeline de validación

FASE	HERRAMIENTAS
Fase de <i>commit</i>	Analizadores sintácticos de Java XML. SchemaFactory de Java. Comprobador de geometría CityGML. Comprobador de semántica CityGML. Generador de estadísticas CityGML.
Fase de <i>3DCityDB - acceptance</i>	3DCityDB - Generador de estructura de bases de datos. 3DCityDB - Importador de CityGML a DB. Desplegador WFS para CityGML. Deegree - Generador de solicitudes WFS. Comprobador de fachadas.
Fase de UAT	El despliegue y las pruebas de humo son similares a los de la fase de <i>acceptance</i> . También se ejecuta la prueba de integración de los elementos de la ciudad en el entorno real. Este paso requiere la intervención del evaluador, de modo que el <i>pipeline</i> quedará en pausa hasta que este aporte alguna respuesta.
Fase de <i>production</i>	Las herramientas empeladas en el entorno de producción son las mismas que las desarrolladas para la fase de aceptación.

El *pipeline* de despliegue se ha implementado con ayuda de Jenkins, una aplicación de código abierto para la integración continua que también es apta para el despliegue continuo [103]. La principal funcionalidad de Jenkins es la ejecución de una serie de pasos en base a un periodo de tiempo o a eventos. Hay un gran número de complementos que amplían su funcionalidad central. En este caso se han utilizado los complementos "Git" y "Solicitud HTTP" ("*HTTP Request*"). El código del archivo de órdenes de Jenkins que se ha usado en este ejemplo está disponible íntegramente en <https://github.com/Tecnalia-CityGML/CityGML-Deployment-Pipeline-Script>. Los servicios recogidos en la Tabla 4-4 han sido desarrollados para este experimento a base de incrustar los datos de configuración del entorno, de modo que ahora están vinculados a nuestros entornos de prueba y producción. En una implementación real reutilizable, estos datos constituirían un parámetro del servicio.

En lo relativo a la fase de UAT, el CityGML se descarga desde el servidor de prueba y se visualiza con Cesium. En la Figura 4-5 se ve la interfaz gráfica de usuario que Jenkins

muestra mientras espera la interacción del evaluador, y en la Figura 4-5 el modelo CityGML tal y como se ve en Cesium.

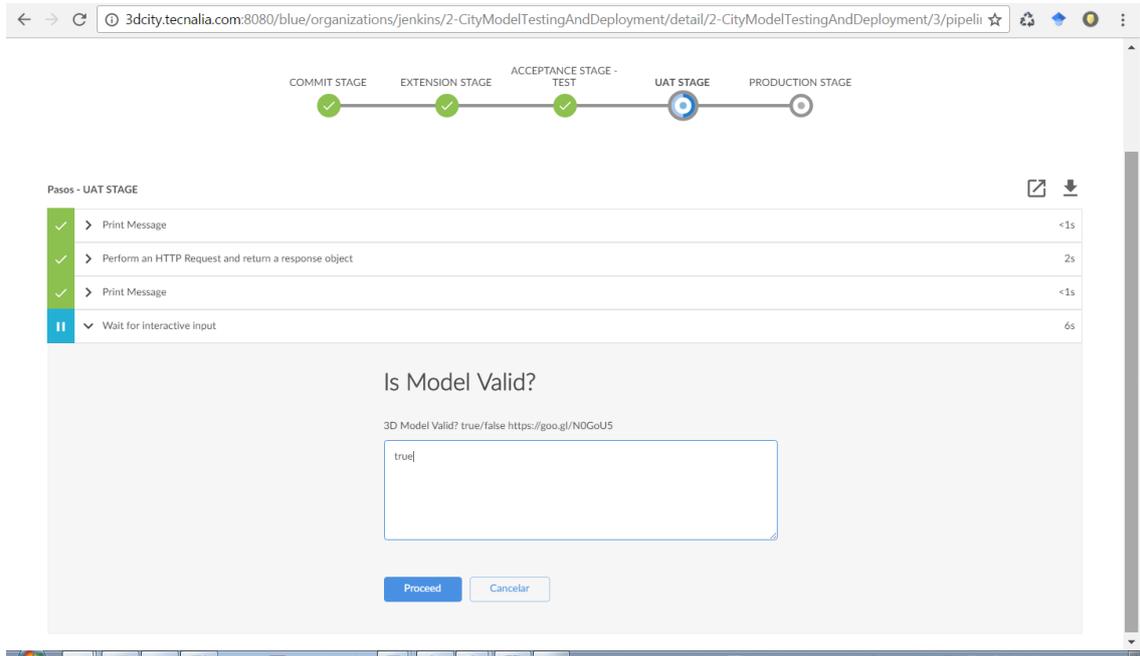


Figura 4-4 Visualización de UAT en Jenkins

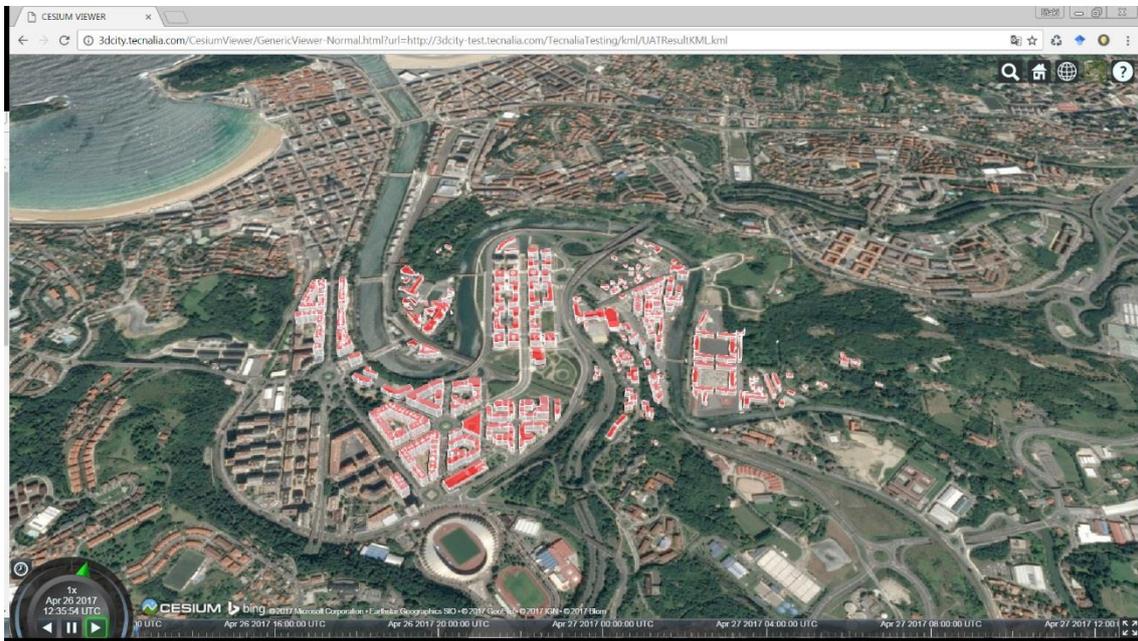


Figura 4-5 Visualización de UAT en Cesium

4.5.2 Ejecución y resultados

En este escenario, los tres participantes introducen ciertos cambios en el modelo. Al actualizar el modelo CityGML con información geométrica y/o semántica, lo someten a comprobaciones a través de sus *pipelines* locales. Una vez que el modelo atraviesa los *pipelines* con éxito, las modificaciones introducidas son enviadas al repositorio compartido y se activa el *pipeline* completo. En el siguiente escenario de muestra se realizan 5 despliegues satisfactorios del modelo de ciudad y se detectan automáticamente otros 5 errores antes de que llegue a los usuarios finales.

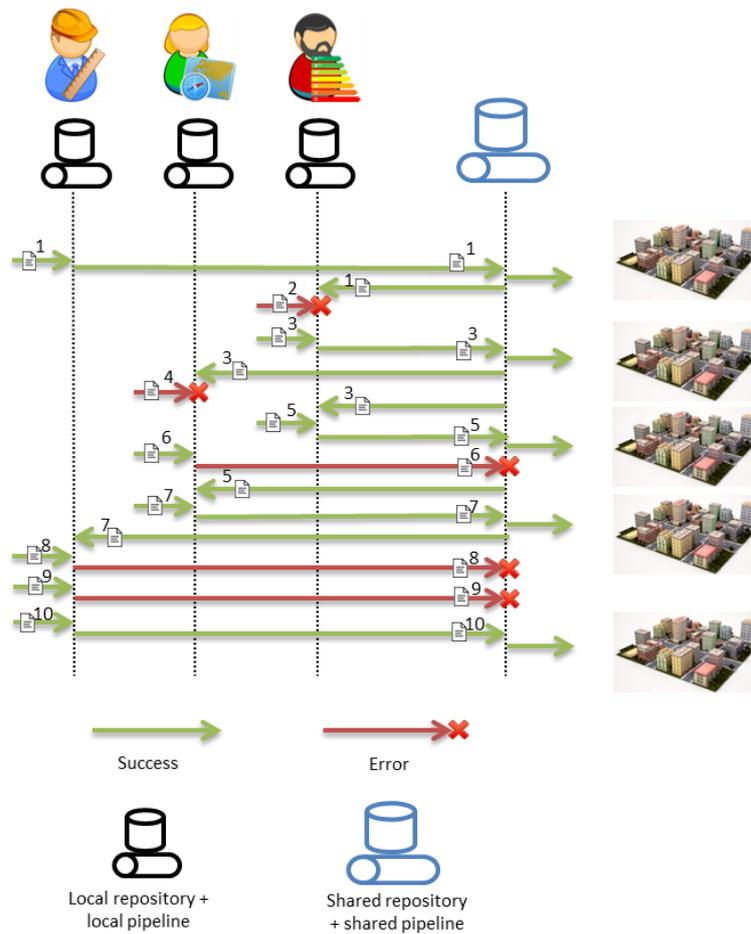


Figura 4-6 Interacciones principales en el escenario de muestra

La Figura 4-6 detalla la secuencia de acciones ejecutadas por todos los participantes en el escenario del experimento:

- En primer lugar, el arquitecto municipal crea el modelo de ciudad con 300 edificios en LOD2 (1). Los *pipelines* locales y compartidos se han completado con éxito, pero se ha advertido de que hay fachadas con área menor a 1m² que posiblemente haya que corregir.
- El consultor de energía extrae el modelo de ciudad (1) y añade la información semántica a 2 edificios (2). Su *pipeline* local fracasa porque los tipos de código de función de los edificios no son correctos. Tras corregirlos (3), tanto el *pipeline* local como el compartido se completan satisfactoriamente.
- Unos días más tarde, la técnica de GIS extrae el modelo de ciudad (3) y completa los 300 edificios, incluidos sus LOD0 y LOD1 (4). Sin embargo, su *pipeline* local no concluye con éxito dado que algunas geometrías contienen errores. La técnica GIS empieza entonces a corregir esos errores geométricos. Mientras tanto, el consultor de energía extrae el modelo de ciudad (3) y completa la información semántica de otro edificio (5). Los *pipelines* local y compartido se completan correctamente. Después de eliminar los fallos en las geometrías (6), el *pipeline* local de la técnica de GIS por fin se completa satisfactoriamente. Pero, al intentar enviar el modelo de ciudad de vuelta al repositorio compartido, se produce un conflicto con otro conjunto de modificaciones enviadas entre medias (5), lo que significa que la técnica de GIS ha estado trabajando localmente sobre una versión obsoleta. Extrae el último modelo de ciudad (5), lo fusiona con sus últimos cambios en su ordenador (7) y, con ello, ambos *pipelines* se completan sin problemas.
- Después, el arquitecto municipal extrae el modelo de ciudad (7) y lo actualiza con los LOD0, LOD1 y LOD2 de 263 edificios (8). El *pipeline* compartido fracasa porque uno de los requisitos del usuario (verificado mediante una prueba de aceptación) exigen que todas las fachadas presenten área y orientación principal. El arquitecto corrige entonces los errores (9) y envía las modificaciones al repositorio compartido, desencadenando así el *pipeline* compartido. En la fase de UAT de ese mismo *pipeline*, un evaluador detecta que algunos edificios no están bien ubicados, de modo que el *pipeline* no llega a completarse. Tras solventar los errores (10), ambos *pipelines* son un éxito.

El repositorio Git empleado en este escenario de muestra está alojado en Github, en <https://github.com/Tecalia-CityGML/CityGML-Deployment-Pipeline.git>. El registro de *commits* que aparece en la Figura 4-7, a la izquierda, se ha tomado de la rama master de dicho repositorio. A la derecha se pueden ver los diferentes desencadenantes del *pipeline* en Jenkins.

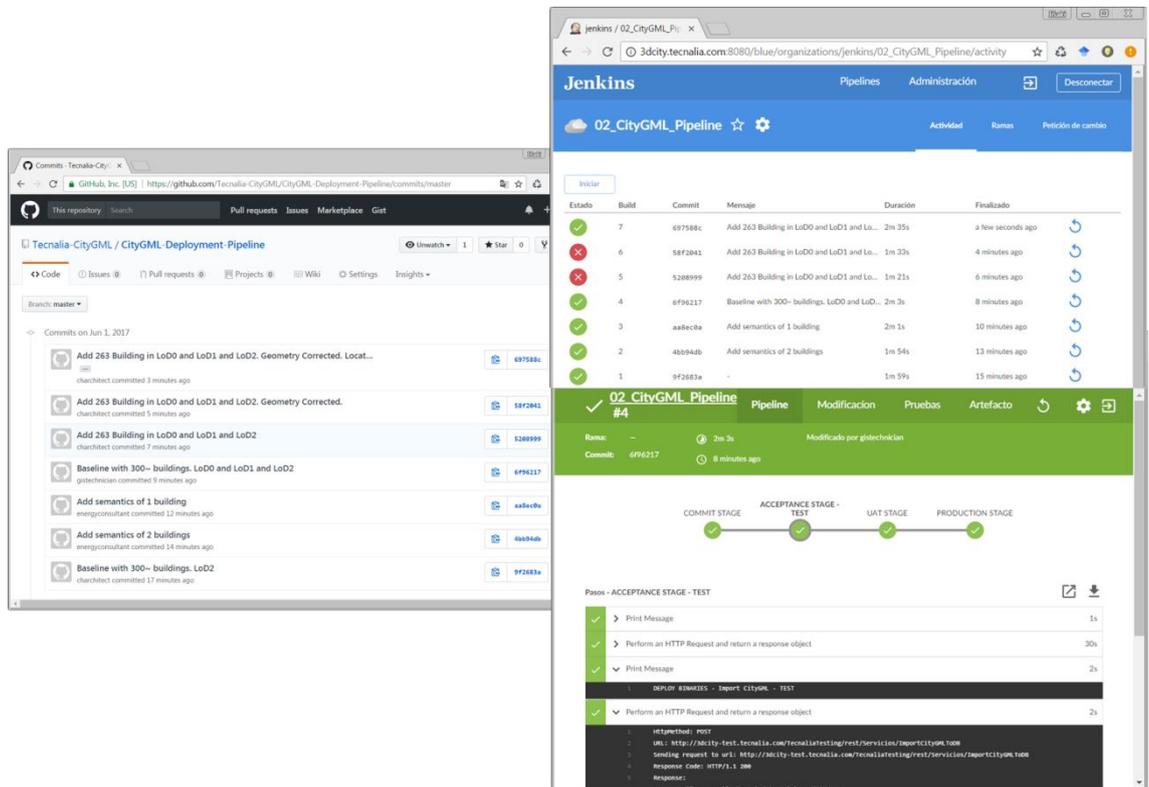


Figura 4-7 Diferentes desencadenantes del pipeline en Jenkins

4.6 Discusión

En este apartado se abordan los motivos detrás de algunas de las decisiones más importantes que se han tomado para desarrollar la propuesta presentada en este capítulo.

En primer lugar, cabe destacar que se ha seleccionado CityGML como modelo de datos porque, tras crear numerosos modelos de ciudades para otros proyectos, se ha podido identificar un problema con las tareas de mantenimiento y validación de los modelos. El estándar CityGML nació hace 10 años, pese a lo que aún sigue habiendo un elevado número de modelos CityGML públicamente disponibles con muchos errores semánticos y geométricos. Sea como sea, los *pipelines* de despliegue basados en un VCS y las pruebas

automáticas y manuales se pueden aplicar a otros formatos de datos complejos, de modo que son opciones para explorar en el futuro.

Actualmente, los modelos se suelen validar manualmente. Dado que se trata de un proceso largo, muchas veces acaba siendo omitido. Tras haber identificado los procesos manuales como una de las principales fuentes de problemas de cara al despliegue y mantenimiento de modelos CityGML válidos, se ha decidido aplicar una estrategia para automatizarlos. En lugar de diseñar e implementar una solución *ad hoc*, se ha recurrido a la estrategia y herramientas que brinda la ingeniería de *software* para el despliegue continuo. La mayor ventaja de adoptar esta estrategia es que ha sido posible beneficiarse de la existencia de potentes herramientas especializadas. Otra ventaja es que, dado que esta estrategia está ganando cada vez más adeptos para el desarrollo de *software*, es más fácil reclutar personas para el mantenimiento y el despliegue de los modelos CityGML con conocimientos y experiencia previos.

La solución propuesta incluye el despliegue de unos modelos CityGML válidos, refiriéndonos por "desplegar" a "poner los modelos a disposición de sus usuarios potenciales". Actualmente no hay demasiados modelos CityGML disponibles y, generalmente, su despliegue es bastante simple (p. ej. se ponen a disposición a través de un servidor web), por lo que su automatización puede parecer innecesaria. No obstante, esta propuesta pretende simplificar y agilizar la introducción de modificaciones en los modelos CityGML. Si se facilitan las modificaciones, se harían cambios con mayor frecuencia. Pero si los cambios son más frecuentes, también lo serán los despliegues, incluso los más sencillos, de modo que consumirán más tiempo. Por eso tiene más sentido automatizar este proceso y no realizarlo de forma manual. Además, los escenarios más complejos (como despliegues en la nube con balance de carga dinámico o despliegues en los que el *software* y los modelos CityGML se empaquetan juntos) se beneficiarían aún más de la automatización de los despliegues.

Para configurar una estrategia de despliegue continuo se necesita un sistema de control de versiones. Las herramientas de despliegue continuo se integran muy bien con VCS y permiten disponer de un registro de modificación, con información sobre quién cambió qué, cuándo y para qué, así como restaurar cualquier versión previa. Además, se requiere algún tipo de control de versiones para facilitar el trabajo colaborativo, aunque en este

caso no tiene por qué ser necesariamente un VCS y se puede insertar directamente dentro del modelo de datos.

Una última ventaja de la propuesta presentada es que se puede adoptar de manera incremental, primero con un repositorio VCS, después con un *pipeline* simple con una sola fase de *commit*, en tercer lugar, con un *pipeline* ampliado con fase de *acceptance* y, finalmente, con un *pipeline* completo. Cada uno de estos pasos son útiles por sí mismos, es decir, que su implementación individual sirve para resolver algunos de los problemas detectados, aunque se deben adoptar en ese orden, porque los últimos no pueden funcionar sin los primeros. La adopción incremental contribuye a eliminar riesgos a la hora de probar esta solución en un entorno real.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

Se presentan a continuación las principales conclusiones extraídas del desarrollo de esta tesis. Se presentan en primer lugar las conclusiones del trabajo descrito en el capítulo 3 y a continuación las del capítulo 4.

5.1.1 Metodología de generación y actualización semiautomática de modelos CityGML

En este trabajo se ha propuesto un flujo de trabajo para la creación de modelos 3D urbanos (generación y mantenimiento) a partir de datos generalmente gratuitos y de naturaleza heterogénea, como solución para poder crear modelos tridimensionales de forma semiautomática con datos tanto geométricos como semánticos (ver Capítulo 3). Este flujo de trabajo engloba el diseño del modelo de datos, el volcado de información geométrica y semántica (incluida la derivación automática de propiedades semánticas) y la a menudo olvidada fase de mantenimiento.

El desarrollo de modelos 3D urbanos en base al estándar CityGML de OGC permite integrar escalas a nivel de ciudad y de edificio en un mismo modelo con datos geométricos y semánticos añadidos. Un modelo de estas características se puede utilizar en numerosas aplicaciones de las que diversos agentes, ya sean planificadores urbanos o ciudadanos, entre otros, podrían beneficiarse.

La creación de los modelos CityGML utilizando el flujo de trabajo de ciclo de vida de ciudades en 3D se divide en tres fases bien definidas. Aunque se ha utilizado para generar los tres modelos CityGML de Santiago, Frankfurt y Budapest (descritos en la sección 3.5), se define de manera general para facilitar la definición o adaptación del modelo a cualquier otro requisito específico. La adaptación se realiza en la fase inicial a través de la configuración del modelo de datos. El flujo de trabajo de generación también se puede adaptar a diferentes fuentes de datos y se centra en el uso de datos abiertos y existentes.

La lista completa de datos requeridos para la generación de los modelos CityGML se realiza una vez y es aplicable a los tres casos de estudio. Los datos requeridos son los mismos para cada caso de estudio, sin embargo, las fuentes de datos pueden ser diferentes. El tamaño del distrito también puede variar para la generación del modelo de ciudad. El tamaño de los casos de estudios varía desde 16 edificios en el distrito y 3.075 edificios en

el área de 2km en el caso de Budapest, a 92 edificios en el distrito y 6.525 edificios en el área de 2km en el caso de Santiago.

Diferentes niveles de detalle han sido generados en CityGML. La geometría de los edificios se representa con diferentes niveles de detalle (LOD0, LOD1 y LOD2) aunque para otros elementos de la ciudad solo se genera LOD1.

Después de la generación geométrica del modelo de ciudad, se han lanzado varios procesos automáticos para completar la información semántica del modelo de datos. Se realiza una semantización (asignación de información semántica a los objetos urbanos) masiva para las propiedades de los edificios en el distrito. El proceso automático completa el modelo de datos generado con las propiedades recopiladas en un archivo SHP. La función del edificio es uno de los parámetros clave del edificio y la información también puede introducirse automáticamente en el modelo de datos de las capas temáticas existentes en formato GIS. También se ha lanzado el procesado geométrico para la semantización del modelo CityGML a través de su geometría.

Es muy habitual que los modelos 3D urbanos creados acaben quedando desactualizados. Sin embargo, a fin de poder hacer un uso realista de los modelos de datos, es imprescindible realizar algunas tareas para mantenerlos "vivos". Por eso, además de las acciones ya presentadas en el flujo de trabajo de generación que pueden repetirse durante el mantenimiento, se han desarrollado otras funcionalidades específicas de dicha etapa del ciclo de vida de un modelo CityGML, tales como: concatenar varios modelos CityGML, reemplazar un edificio por otro en un modelo CityGML, seleccionar un conjunto de elementos de una zona del modelo CityGML o completar el modelo CityGML con un modelo de un edificio más detallado, entre otros.

Por último, se ha descrito el componente “Tecnalia_CityGML_Processing”, que implementa un conjunto de servicios web diseñados para implementar distintos elementos del flujo de trabajo descritos en la sección 3.3.

5.1.2 Validación y despliegue colaborativo de modelos CityGML

En este trabajo también se ha propuesto un método para facilitar el despliegue y mantenimiento regular de unos modelos de ciudad 3D sin errores en CityGML (ver Capítulo 4). Este método consiste en diseñar, implementar y aplicar un *pipeline* de despliegue para cada modelo CityGML que deba ser creado, desplegado y mantenido.

Los *pipelines* de despliegue CityGML aplican pues los procedimientos y comprobaciones necesarios para transformar un conjunto de modificaciones en un modelo CityGML validado, actualizado y desplegado o para arrojar información sobre los problemas detectados.

Se ha desarrollado y aplicado un *pipeline* de despliegue CityGML en el marco de un escenario de prueba que, por un lado, representa los tipos de problemas que esta propuesta pretende resolver y, por otro, está basado en trabajos que se están llevando a cabo en la realidad. Cabe destacar que algunas de las herramientas CityGML específicas que se han implementado en este ejemplo tienen los datos de configuración del entorno de despliegue incrustados, de manera que ahora mismo son prototipos específicos para los entornos de prueba y producción del experimento. En una implementación reutilizable de estas herramientas, los datos de configuración se deberían aportar en forma de parámetros.

La estructura de los *pipelines* de despliegue de CityGML se basa en la de los *pipelines* de despliegue continuo empleados en el despliegue automático de *software*, aunque se han modificado para abordar un problema distinto. Una vez que la estructura genérica está definida, se debe proceder a diseñar e implementar un *pipeline* específico para cada modelo CityGML. El diseño seleccionará (o añadirá) fases y tareas en función de las restricciones y necesidades de cada caso, estando algunas relacionadas con el dominio del problema que el modelo CityGML pretende solucionar. Con su diseño completo, los *pipelines* se pueden implementar en cualquier lenguaje de programación, aunque se recomienda utilizar herramientas para el control de versiones y el despliegue continuo (como Jenkins y Git, que son las empleadas en el escenario de muestra presentado), porque están concebidas específicamente para resolver este tipo de problemas.

Uno de los inconvenientes del estándar CityGML es que, tal y como se desprende de las investigaciones más recientes, actualmente parece difícil crear modelos totalmente correctos [82]. Esta solución aspira a simplificar, mantener y desplegar modelos CityGML validados. Se espera contribuir a mejorar la calidad y disponibilidad de los modelos CityGML y a fomentar el uso de este estándar.

5.2 Principales contribuciones

A continuación, se resumen las conclusiones de las principales contribuciones del trabajo realizado en esta investigación:

(1) Diseño y desarrollo de una solución para el mantenimiento colaborativo y despliegue automático de modelos CityGML validados.

Se ha presentado una solución que facilita el mantenimiento regular modelos 3D urbanos en CityGML. Para ello, la solución se fundamenta en la estrategia y las herramientas de despliegue continuo empleadas en el desarrollo de *software* y se ha adaptado al problema derivado de crear, mantener y desplegar modelos CityGML validados. Ninguna de las herramientas existentes permite validar y desplegar el modelo 3D urbano cada vez que se realiza un cambio.

La solución propuesta en esta tesis describe un método para automatizar el proceso de validación y despliegue de modelos CityGML basado en dos pasos: 1) Definir una secuencia de pasos automáticos vinculados entre sí (*pipeline*), para transformar un modelo CityGML alojado en un repositorio VCS, en un modelo CityGML validado y desplegado. 2) Ejecutar el pipeline en cuestión cada vez que se realiza alguna modificación en un modelo CityGML y se actualiza en el VCS. El pipeline sigue la misma estructura de fases que los pipelines de despliegue continuo de software y se aprovecha de herramientas utilizadas para el software, ya que hay gran variedad y están muy extendidas y bien documentadas. Adicionalmente, se han desarrollado herramientas y servicios específicos para el pipeline de despliegue de modelos CityGML y se han utilizado librerías de terceros específicas para estos modelos.

La validación se ha realizado desarrollando y aplicando un pipeline de despliegue CityGML en el marco de un escenario de muestra que representa los tipos de problemas que esta propuesta pretende resolver y, al mismo tiempo está basado en un flujo de trabajo real. Se ha configurado un entorno de prueba con el equipamiento y las herramientas necesarias para probar la solución propuesta en el escenario de ejemplo. Se han ejecutado todas las fases del pipeline y se han realizado varios despliegues satisfactorios del modelo CityGML.

(2) Diseño y desarrollo de una metodología de generación y actualización semiautomática de modelos CityGML.

Se ha descrito un flujo de trabajo que define las actividades a realizar para generar un modelo 3D urbano que incluya información geométrica y semántica de edificios y otros elementos urbanos en diferentes niveles de detalle. Actualmente la mayoría de las herramientas de generación de modelos 3D urbanos se centran únicamente en los edificios, sin tener en cuenta el resto de elementos de la ciudad. Tampoco se tienen en cuenta las necesidades del usuario desde el comienzo. El flujo de trabajo propuesto en esta tesis permite diseñar las extensiones de dominio desde la generación del modelo. Por otra parte, la semantización del modelo se suele realizar una única vez, mientras que en esta investigación se tiene en cuenta cómo realizar las operaciones de mantenimiento del modelo 3D urbano.

Se han descrito también una serie de procesos que permiten realizar operaciones de mantenimiento sobre el modelo 3D urbano con el fin de permitir su actualización. Este flujo de trabajo engloba el diseño del modelo de datos, el volcado de información geométrica y semántica (incluida la derivación automática de propiedades semánticas) y la a menudo olvidada fase de mantenimiento.

También se han desarrollado un conjunto de servicios web para automatizar varios pasos del flujo de trabajo descrito, tanto para la generación como para las operaciones de mantenimiento. Tanto el flujo de trabajo descrito como los servicios desarrollados se han utilizado en la implementación de los tres modelos CityGML parciales de tres ciudades: Santiago de Compostela, Frankfurt y Budapest.

(3) Presentación de tres casos de estudio que motivan el uso de los modelos 3D urbanos.

Se han descrito y analizado tres casos de estudio que han permitido aprovechar el potencial del modelo 3D urbano en herramientas con diferentes finalidades: (1) categorizar el parque edificado de una ciudad, (2) realizar la gestión colaborativa para la definición y seguimiento de un PERI y (3) evaluar el impacto de aplicación de medidas de conservación energética en entornos urbanos históricos.

Los casos de estudio son un ejemplo que reflejan la utilidad de los modelos 3D urbanos. La necesidad de disponer de estos modelos va en aumento al igual que su utilización en diferentes aplicaciones.

Los resultados obtenidos del análisis de estos casos de estudio han permitido extraer conclusiones que motivan el desarrollo de tecnologías y herramientas para la automatización de procesos de generación, mantenimiento, testeo y despliegue de modelos CityGML.

(4) **Identificación de requisitos de los modelos 3D urbanos.**

Se han definido los requisitos del modelo 3D urbano analizando los tres casos de estudio presentados y analizando herramientas existentes que permiten generar, editar y visualizar modelos 3D urbanos. De esta forma se han definido algunos de los requisitos que debe cumplir un modelo 3D urbano para que sea reutilizable en diferentes ámbitos y aplicaciones.

Los requisitos se presentan estructurados por categorías que representan cada una de las principales etapas del ciclo de vida de un modelo 3D urbano. Adicionalmente se incluyen requisitos generales asociados a varias etapas o independientes de las mismas. Estos requisitos se traducen en la definición de un modelo de datos común, multiescala, genérico, interoperable y que contiene la información semántica y geométrica necesaria para la gestión, la toma de decisiones para la mejora de la sostenibilidad y su posterior gestión y mantenimiento.

5.3 Trabajo futuro

A continuación, se presentan algunas ideas de líneas futuras de continuación del trabajo realizado.

Los resultados descritos en el capítulo 3 abren la puerta a varias posibilidades de trabajo futuro. En primer lugar, el flujo de trabajo incluye una serie de acciones manuales, de modo que sigue siendo necesario estudiar otras opciones para automatizarlas al máximo. Un ejemplo consiste en enlazar automáticamente el proceso de generación con alguna ADE ya existente, tales como EnergyADE o UtilityNetworkADE. Cabe destacar también que el mantenimiento a largo plazo de los modelos de ciudad 3D sigue representando un desafío. Para poder comprobar si la propuesta presentada hace la tarea de mantenimiento

más factible a un coste razonable no solo deberemos llevar a cabo pruebas de larga duración, sino, sin duda, introducir algunas mejoras para poder hacer frente a los problemas que se vayan descubriendo. Para ello es posible diseñar y desarrollar nuevas operaciones de mantenimiento, como por ejemplo la actualización automática de información semántica a partir de datos abiertos.

Los resultados descritos en el capítulo 4 permiten probar (y adaptar) este enfoque al despliegue continuo de modelos de ciudad 3D en escenarios más complejos. El reto a futuro tiene que ser automatizar todo lo automatizable (como las tareas que actualmente son manuales en el UAT). Para ello es necesario ir aumentando el número de validaciones añadiendo nuevas, como por ejemplo: validar la consistencia geométrico-semántica del modelo. El uso de *pipelines* de despliegue CityGML debería facilitar la configuración de escenarios de mantenimiento CityGML más amplios y colaborativos en los cuales se dé un salto importante en el número de usuarios, tamaño del modelo CityGML y número de actualizaciones realizadas. Por ejemplo, se podría crear un escenario de una ciudad en el que los diferentes departamentos del Ayuntamiento participasen en el despliegue continuo del modelo urbano. Se podría permitir que empresas externas, o incluso ciudadanos voluntarios, contribuyesen con actualizaciones y correcciones del modelo.

El urbanismo participativo [104] es otra de las áreas de aplicación en las que esta propuesta puede ser de provecho. Por ejemplo, se podría abrir una convocatoria pública para proponer el diseño de un nuevo parque basado en el modelo CityGML común de la ciudad. Se podrían presentar, comparar y combinar varias alternativas en función de la flexibilidad que brinde un DVCS y de la red de seguridad contra errores resultante de las pruebas que se ejecuten en los *pipelines* de despliegue. Este tipo de entornos exigirá cierto grado de experimentación para descubrir qué flujos de trabajo, herramientas y prácticas son las mejores.

Otra línea de investigación futura que está ligada con los resultados de este trabajo es la integración de modelos y herramientas que permiten la representación de la información detallada de los elementos urbanos a diferentes escalas (integración CityGML-BIM), la combinación de información espacial y temporal (series de datos temporales) y la integración con herramientas de simulación. Esta integración permitirá mejorar la precisión y resultados cuantitativos en aplicaciones de gestión y toma de decisiones en los procesos urbanos.

6 Listado de méritos

6.1 Publicaciones del capítulo 2

Egusquiza, A., Prieto, I., Izkara, J. L., & Béjar, R. (2018). Multi-scale urban data models for early-stage suitability assessment of energy conservation measures in historic urban areas. *Energy and Buildings*.

Gandini, A., Prieto, I., Garmendia, L., San-Jose, J. T., & Egusquiza, A. (2018). Adaptation to flooding events through vulnerability mapping in historic urban areas. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives*.

Prieto, I., Izkara, J.L., Mediavilla, A., Arambarri, J. & Arroyo, A. (2018). Collaborative platform based on standard services for the semi-automated generation of the 3D city model on the cloud. *European Conference on Product and Process Modelling. ECPPM 2018*. Aceptada para su presentación en el congreso y su posterior publicación.

Prieto, I., Izkara, J. L., & Egusquiza, A. (2017). Building stock categorization for energy retrofitting of historic districts based on a 3D city model. *DYNA-Ingeniería e Industria*, 92(5).

Prieto I., Izkara J.L., Béjar R. (2017) Web-Based Tool for the Sustainable Refurbishment in Historic Districts Based on 3D City Model. In: Abdul-Rahman A. (eds) *Advances in 3D Geoinformation. Lecture Notes in Geoinformation and Cartography*. Springer, Cham.

Prieto, I., Izkara, J. L., Egusquiza, A., & Béjar, R. (2015). Sustainable refurbishment in urban districts through a web-based tool based on 3D city model. *Sustainable Places*, 2015, 31.

Egusquiza, A., Prieto, I., & Romero, A. (2014). Multiscale information management for sustainable districts rehabilitation: EFFESUS and FASUDIR projects. *eWork and eBusiness in Architecture, Engineering and Construction ECPPM*, 303-308.

6.2 Publicaciones del capítulo 3

Prieto, I., Izkara, J. L., & Bejar, R. (2014). A workflow for the semi-automatic generation of low cost 3D city models. *LowCost3D - Sensors, Algorithms, Applications*.

Prieto, I., Usobiaga, E., & Izkara, J. L. (2012). Generación semiautomática de ciudades en 3D en CityGML a partir de fuentes de datos libres. X Congreso TOPCART 2012 - I Congreso Iberoamericano de Geomática Y C.C. de La Tierra.

Mediavilla, A., Izkara, J. L., & Prieto, I. (2015). HOLISTEEC—Plataforma colaborativa en la nube basada en BIM para el diseño de edificios energéticamente eficientes. *Spanish Journal of Building Information Modeling*, 15(1), 4-11.

6.3 Publicaciones del capítulo 4

Prieto, I., Izkara, J. L., & Béjar, R. (2018). A continuous deployment-based approach for the collaborative creation, maintenance, testing and deployment of CityGML models. *International Journal of Geographical Information Science*, 32(2), 282-301.

7 Referencias

- [1] U. Europea, “Las ciudades del mañana. Retos, visiones y caminos a seguir,” *Com. Eur. Dir. Gen. Política Reg. Bruselas*, 2011.
- [2] OECD/IEA, “International Energy Agency,” 2013. [Online]. Available: <http://www.iea.org/aboutus/faqs/energyefficiency/>.
- [3] D. U. N. P. of Economic, *World urbanization prospects: The 2009 revision. Division, Social Affairs*. UN, 2010.
- [4] A. Ferretti, “An Indicator-based Approach to Measuring Regeneration of Historic Cities,” *Ital. J. Plan. Pract.*, vol. 4, no. 1, pp. 121–156, 2014.
- [5] F. J. Delgado, R. Martínez, I. Prieto, J. L. Izgara, A. Egusquiza, and J. Finat, “A common framework for multidisciplinary information management in historic urban districts,” in *EuroGEOSS 2012 Conference. Madrid*, 2012.
- [6] G. de España, “PLAN ESTATAL DE INVESTIGACIÓN CIENTÍFICA Y TÉCNICA Y DE INNOVACIÓN. 2017 - 2020,” *Ministerio de Economía, Industria y Competitividad*, 2017. [Online]. Available: <http://www.idi.mineco.gob.es/stfls/MICINN/Prensa/FICHEROS/2018/PlanEstateIID.pdf>.
- [7] European Commission, “Smart Cities and Communities. The European Innovation Partnership on Smart Cities and Communities,” 2017. [Online]. Available: <http://ec.europa.eu/eip/smartcities/>.
- [8] I. Directive, “Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE),” *Publ. Off. J. 25th April*, 2007.
- [9] L. Ross, J. Bolling, J. Döllner, and B. Kleinschmit, “Enhancing 3d city models with heterogeneous spatial information: Towards 3d land information systems,” *Adv. GIScience*, pp. 113–133, 2009.
- [10] L. van den Brink, J. Stoter, and S. Zlatanova, “Establishing a national standard for 3D topographic data compliant to CityGML,” *Int. J. Geogr. Inf. Sci.*, vol. 27, no. 1, pp. 92–113, 2012.
- [11] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, “Applications of 3D City Models: State of the Art Review,” *ISPRS Int. J. Geo-Information*, vol. 4, no. 4, pp. 2842–2889, 2015.
- [12] M. Aktaruzzaman, T. G. Schmitt, and H. Hagen, “Modeling Urban Flooding by Filtering LiDAR Data,” *J. Urban Technol.*, vol. 18, no. 4, pp. 97–112, 2011.
- [13] T. Krüger and T. Kolbe, “Building analysis for urban energy planning using key indicators on virtual 3D city models—the energy atlas of Berlin,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. Vol. XXXIX-B2, 2012. XXII ISPRS Congr. 25 August – 01 Sept. 2012, Melbourne, Aust.*, vol. XXXIX, pp. 145–150, 2012.

- [14] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manag. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.
- [15] R. E. Stake, *Investigación con estudio de casos*. Ediciones Morata, 1998.
- [16] D. E. Stokes, *Pasteur's quadrant: Basic science and technological innovation*. Brookings Institution Press, 2011.
- [17] D. Arnold, Ed., "Editorial for inaugural issue of {JOCCH}," *J. Comput. Cult. Herit.*, vol. 1, no. 1, pp. 1–13, Jun. 2008.
- [18] P. J. Morton, M. Horne, R. C. Dalton, and E. M. Thompson, "Virtual City Models: avoidance of obsolescence," *Digit. Phys. 30th eCAADe Confer-Comparative Eval. two Methodol.*, 2012.
- [19] B. Mao and Y. Ban, "Online Visualization of 3D City Model Using CityGML and X3DOM," *Cartogr. Int. J. Geogr. Inf. Geovisualization*, vol. 46, no. 2, pp. 109–114, 2011.
- [20] R. Chen, "The development of 3D city model and its applications in urban planning," *19th Int. Conf. Geoinformatics*, pp. 1–5, 2011.
- [21] G. Gröger and L. Plümer, "CityGML – Interoperable semantic 3D city models," *ISPRS. J. Photogramm. Remote Sens.*, vol. 71, no. 0, pp. 12–33, 2012.
- [22] R. Nouvel, M. Zirak, H. Dastageeri, V. Coors, and U. Eicker, "Urban Energy Analysis Based on 3D City Model for National Scale Applications," in *Presented at the IBPSA Germany Conference*, 2014, vol. 8.
- [23] R. Kaden and T. H. Kolbe, "City-wide total energy demand estimation of buildings using semantic 3d city models and statistical data," in *ISPRS Annals. Presented at the ISPRS 8th 3DGeoInfo Conference, Istanbul, Turkey*, 2013, pp. 163–171.
- [24] A. Henn, C. Römer, G. Gröger, and L. Plümer, "Automatic classification of building types in 3D city models," *Geoinformatica*, vol. 16, no. 2, pp. 281–306, 2012.
- [25] L. Heischbourg, S. Guhathakurta, H. Hagen, S. Petsch, and K. Müller, "Modeling, Monitoring, and Visualizing Carbon Footprints at the Urban Neighborhood Scale," *J. Urban Technol.*, vol. 18, no. 4, pp. 81–96, 2011.
- [26] H. M. Chen, C. C. Hou, and Y. H. Wang, "A 3D visualized expert system for maintenance and management of existing building facilities using reliability-based method," *Expert Syst. Appl.*, vol. 40, pp. 287–299, 2013.
- [27] V. Steinhage, J. Behley, S. Meisel, and A. B. Cremers, "Reconstruction by components for automated updating of 3D city models," *Appl. Geomatics*, vol. 5, no. 4, pp. 285–298, 2013.
- [28] R. N. R. Kuchta, "Smart City Concept, Applications and Services," *J. Telecommun. Syst. Manag.*, vol. 3, no. 2, 2014.
- [29] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering

- algorithm,” *J. R. Stat. Soc. Ser. C (Applied Stat.,* vol. 28, no. 1, pp. 100–108, 1979.
- [30] A. Egusquiza, “Multiscale information management for historic districts’ energy retrofitting : a framework, a methodology, a model,” Universitat Politècnica de Catalunya, 2015.
- [31] T. Grey, M. Dyer, and D. Gleeson, “Using Big and Small Urban Data for Collaborative Urbanism,” in *Citizen Empowerment and Innovation in the Data-Rich City*, Springer, 2017, pp. 31–54.
- [32] A. Goldberg, K. M. Leyden, and T. J. Scotto, “Untangling what makes cities liveable: happiness in five cities,” *Proc. Inst. Civ. Eng. - Urban Des. Plan.,* vol. 165, no. 3, pp. 127–136, Sep. 2012.
- [33] D. Bradley, J. Bradley, M. Coombes, and E. Tranos, “Sense of Place and Social Capital and the Historic Built Environment,” *CURDS/Bradley Res. Consult. English Herit.,* 2009.
- [34] R. Europy, “European Charter of the Architectural Heritage.” Strasburg, 1975.
- [35] M. Economidou, B. Atanasiu, C. Despret, J. Maio, I. Nolte, O. Rapf, and others, “Europe’s buildings under the microscope,” *A Country-by-country Rev. energy Perform. Build.,* vol. 131, 2011.
- [36] L. Mazzarella, “Energy retrofit of historic and existing buildings. The legislative and regulatory point of view,” *Energy Build.,* vol. 95, pp. 23–31, May 2015.
- [37] K. K. Y. Wong, “Economic Value of 3D Geographic Information,” *EuroSDR Dep. Comput. Sci. Univ. Coll. London London, UK,* 2015.
- [38] G. Gröger, T. H. Kolbe, C. Nagel, and K.-H. Häfele, *OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Version 2.0.0.* 2012.
- [39] N. Vandysheva, V. Tikhonov, P. Van Oosterom, J. Stoter, H. Ploeger, R. Wouters, and V. Penkov, *3D Cadastre modelling in Russia.* International Federation of Surveyors (FIG); Ordre National des Ingénieurs Géomètres Topographes (ONIGT), 2011.
- [40] R. Nouvel, C. Schulte, U. Eicker, D. Pietruschka, and V. Coors, “CityGML-based 3D city model for energy diagnostics and urban energy policy support,” *IBPSA World,* vol. 2013, pp. 1–7, 2013.
- [41] Y. Ham and M. Golparvar-Fard, “Mapping actual thermal properties to building elements in gbXML-based BIM for reliable building energy performance modeling,” *Autom. Constr.,* vol. 49, pp. 214–224, 2015.
- [42] S. Cox, A. Cuthbert, P. Daisey, J. Davidson, S. Johnson, E. Keighan, R. Lake, M. Mabrouk, S. Margoulies, R. Martell, and others, “OpenGIS@Geography Markup Language (GML) Implementation Specification, version.” Academic Press, 2002.
- [43] R. Carey and G. Bell, *The annotated VRML 2.0 reference manual.* Addison-Wesley, 1997.
- [44] M. Barnes, E. L. Finch, and Sony Computer Entertainment, “COLLADA – Digital

- Asset Schema Release 1.5.0 Specification,” *Elements*, no. April, 2008.
- [45] Open Geospatial Consortium, “OGC KML 2.3,” p. 266, 2015.
- [46] aecXML.org, “aecXML Preliminary Specification,” 1999.
- [47] N. Nisbet and T. Liebich, “ifcXML implementation guide,” *Int. Alliance Interoperability*, 2005.
- [48] A. Egusquiza, I. Prieto, and A. Romero, “Multiscale information management for sustainable districts rehabilitation EFFESUS and FASUDIR projects,” in *eWork and eBusiness in Architecture, Engineering and Construction*, CRC Press, 2014, pp. 303–308.
- [49] C. Catita, P. Redweik, J. Pereira, and M. C. Brito, “Extending solar potential analysis in buildings to vertical facades,” *Comput. Geosci.*, vol. 66, pp. 1–12, 2014.
- [50] T. H. Kolbe, C. Nagel, and J. Herrerueta, “3D City Database for CityGML,” *Add. to 3D City Database Doc. Version*, vol. 2, no. 1, 2013.
- [51] W. Tegtmeier, S. Zlatanova, P. J. M. van Oosterom, and H. R. G. K. Hack, “3D-GEM: Geo-technical extension towards an integrated 3D information model for infrastructural development,” *Comput. Geosci.*, vol. 64, pp. 126–135, 2014.
- [52] S. A. Aydar, J. Stoter, H. Ledoux, E. D. Ozbek, and T. Yomralioglu, “ESTABLISHING A NATIONAL 3D GEO-DATA MODEL FOR BUILDING DATA COMPLIANT TO CITYGML: CASE OF TURKEY.,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 41, 2016.
- [53] S. Coccolo, D. Mauree, J. Kämpf, and J.-L. Scartezzini, “Integration of outdoor human comfort in a building energy simulation database using CityGML Energy Ade,” in *Expanding Boundaries-Systems Thinking in the Built Environment-Proceedings of the Sustainable Built Environment (SBE) Regional Conference Zurich 2016*, 2016, no. EPFL-CONF-220658.
- [54] R. Béjar, M. Á. Latre, J. Noguera-Iso, P. R. Muro-Medrano, and F. J. Zarazaga-Soria, “An RM-ODP enterprise view for spatial data infrastructures,” *Comput. Stand. Interfaces*, vol. 34, no. 2, pp. 263–272, Feb. 2012.
- [55] J. Döllner, T. H. Kolbe, F. Liecke, T. Sgouros, and K. Teichmann, “The Virtual 3D City Model of Berlin-Managing, Integrating, and Communicating Complex Urban Information,” in *Proceedings of the 25th International Symposium on Urban Data Management UDMS 2006 in Aalborg, Denmark, 15-17 May 2006*, 2006.
- [56] F. Biljecki, H. Ledoux, and J. Stoter, “Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity,” *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. IV-4/W1, pp. 51–59, 2016.
- [57] S. S. Mirvahabi and R. Ali Abaspour, “Extraction of CityGML Building Models in LOD4 Based on Voluntary Data of OSM Database,” *J. Geomatics Sci. Technol.*, vol. 6, no. 1, pp. 185–198, 2016.
- [58] F. Biljecki, H. Ledoux, and J. Stoter, “Generating 3D city models without elevation

- data,” *Comput. Environ. Urban Syst.*, vol. 64, pp. 1–18, 2017.
- [59] H. Ariefi, J. Engels, M. Hahn, and H. Mayer, “Levels of Detail in 3D Building Reconstruction from LiDAR Data,” *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XXXVII, p. 485, 2008.
- [60] L. Malambo and M. Hahn, “LiDAR assisted CityGML creation,” in *3rd Summer Applied Geoinformatics for Society and Environment*, 2010, vol. 211.
- [61] R. Rodríguez, M. Álvarez, M. Miranda, A. Díez, F. Papi, and P. Rodríguez, “Obtención de modelos urbanos tridimensionales,” *Inf. la Construcción*, vol. 65, no. 530, pp. 229–240, 2013.
- [62] G. Navratil, R. Bulbul, and A. U. Frank, “Maintainable city models for sustainable development,” *Int. J. Sustain. Soc.*, vol. 5, no. 2, pp. 97–113, 2013.
- [63] K. Chaturvedi, C. S. Smyth, G. Gesquière, T. Kutzner, and T. H. Kolbe, “Managing versions and history within semantic 3D city models for the next generation of CityGML,” in *Selected papers from the 3D GeoInfo 2015 Conference*, 2015.
- [64] J. Stoter, H. Ledoux, S. Zlatanova, and F. Biljecki, “Towards sustainable and clean 3D Geoinformation,” *Geoinformationssysteme*, pp. 100–113, 2016.
- [65] G. Gröger, T. H. Kolbe, J. Schmittwilken, V. Stroh, and L. Plümer, “Integrating versions, history and levels-of-detail within a 3D geodatabase,” in *Proc. of Int. Workshop on Next Generation City Models, Bonn, EuroSDR publications*, 2005.
- [66] M. Morel and G. Gesquière, “Managing Temporal Change of Cities with CityGML,” *Eurographics Work. Urban Data Model. Vis.*, pp. 37–42, 2014.
- [67] P. Van Oosterom and J. Stoter, “5D data modelling: full integration of 2D/3D space, time and scale dimensions,” in *International Conference on Geographic Information Science*, 2010, pp. 310–324.
- [68] K. Chaturvedi and T. H. Kolbe, “Dynamizers: modeling and implementing dynamic properties for semantic 3d city models,” in *Proceedings of the Eurographics Workshop on Urban Data Modelling and Visualisation*, 2015, pp. 43–48.
- [69] F. Biljecki, H. Ledoux, J. Stoter, and G. Vosselman, “The variants of an LOD of a 3D building model and their influence on spatial analyses,” *ISPRS J. Photogramm. Remote Sens.*, vol. 116, pp. 42–54, 2016.
- [70] J. Benner, A. Geiger, K. Leinemann, F. Karlsruhe, A. Informatik, J. Benner, A. Geiger, and K. Leinemann, “Flexible Generation of Semantic 3D Building Models,” *Building*, vol. 49, pp. 17–22, 2005.
- [71] L. Herman and others, “CityGML - modern approach to city modelling,” *Praha, Digit. Technol. v geoinformatice a Kartogr.*, pp. 17–25, 2011.
- [72] S. Higgins, “The DCC Curation Lifecycle Model,” *Int. J. Digit. Curation*, vol. 3, no. 1, pp. 134–140, 2008.
- [73] D. Bell, “UML basics Part II: The activity diagram,” 2003.

- [74] A. Almohammad, J. F. Ferreira, A. Mendes, and P. White, "ReqCap: Hierarchical Requirements Modeling and Test Generation for Industrial Control Systems," in *2017 {IEEE} 25th International Requirements Engineering Conference Workshops ({REW})*, 2017.
- [75] L. van den Brink, J. E. Stoter, and S. Zlatanova, "Modeling an application domain extension of CityGML in UML," in *Proceedings 3D geoinfo symposium, Quebec, Canada*, 2012, pp. 16–17.
- [76] Y. J. Kim, H. Y. Kang, and J. Lee, "Development of indoor spatial data model using CityGML ADE," *J. ISPRS-International Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, pp. 41–45, 2013.
- [77] I. Prieto, J. L. Izgara, and A. Egusquiza, "Modelo de Información Multiescala Urbana 3D para la gestión integral sostenible de la ciudad," *JIIIDE 2013-IV Jornadas Iber. Infraestructuras Datos Espac.*, 2013.
- [78] W. F. Limp, A. Payne, S. Winters, A. Barnes, and J. Cothren, "Approaching 3D Digital Heritage Data from a Multi-technology, Lifecycle Perspective," in *Proceedings of the 38th Annual International Conference on Computer Applications and Quantitative Methods in Archaeology (CAA), Granada, Spain*, 2010.
- [79] R. Cao, Y. Zhang, X. Liu, and Z. Zhao, "3D building roof reconstruction from airborne LiDAR point clouds: a framework based on a spatial database," *Int. J. Geogr. Inf. Sci.*, pp. 1–22, 2017.
- [80] T. W. Kang and C. H. Hong, "IFC-CityGML LOD Mapping Automation based on Multi-Processing," in *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, 2015, vol. 32, p. 1.
- [81] J. Benner, A. Geiger, K. Leinemann, F. Karlsruhe, A. Informatik, J. Benner, A. Geiger, and K. Leinemann, "Flexible Generation of Semantic 3D Building Models," in *Building*, 2005, vol. 49, pp. 17–22.
- [82] F. Biljecki, H. Ledoux, X. Du, J. Stoter, K. H. Soon, and V. H. S. Khoo, "The Most Common Geometric and Semantic Errors in CityGML Datasets," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, pp. 13–22, 2016.
- [83] R. Peters, J. Stoter, and H. Ledoux, "3D city modelling," in *European Spatial Data Research*, 2014, p. 122.
- [84] N. R. Budhathoki and C. Haythornthwaite, "Motivation for Open Collaboration: Crowd and Community Models and the Case of OpenStreetMap," *Am. Behav. Sci.*, vol. 57, no. 5, pp. 548–575, 2012.
- [85] S. Abbasi and M. R. Malek, "Design and Modeling of a 3D Volunteered Geographic Information with an Interoperable Description for Fundamental Components of a Building," *J. Geomatics Sci. Technol.*, vol. 4, no. 4, pp. 15–28, 2015.
- [86] M. Goetz and A. Zipf, "OpenStreetMap in 3D--Detailed Insights on the Current Situation in Germany," in *City, Proceedings of the AGILE 2012 International Conference on Geographic Information Science, Avignon, April*, 2012, pp. 24–27.

- [87] M. Goetz, "Towards generating highly detailed 3D CityGML models from OpenStreetMap," *Int. J. Geogr. Inf. Sci.*, pp. 1–21, 2012.
- [88] T. Scully, J. Doboš, T. Sturm, and Y. Jung, "3drepo. io: building the next generation Web3D repository with AngularJS and X3DOM," in *Proceedings of the 20th International Conference on 3D Web Technology*, 2015, pp. 235–243.
- [89] M. Sindram, T. Machl, H. Steuer, M. Pültz, and T. H. Kolbe, "Voluminator 2.0 - speeding up the approximation of the volume of defective 3d building models," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. III, no. July, pp. 12–19, 2016.
- [90] F. Biljecki, G. B. M. Heuvelink, H. Ledoux, and J. Stoter, "Propagation of positional error in 3D GIS: estimation of the solar irradiation of building roofs," *Int. J. Geogr. Inf. Sci.*, vol. 29, no. 12, pp. 2269–2294, 2015.
- [91] G. Gröger and V. Coors, "Modeling guide for 3D objects," *SIG 3D*, 2011.
- [92] D. Wagner, M. Wewetzer, J. Bogdahn, N. Alam, M. Pries, and V. Coors, "Geometric-semantic consistency validation of CityGML models," in *Lecture Notes in Geoinformation and Cartography*, 2013, pp. 171–192.
- [93] J. Zhao, J. Stoter, and H. Ledoux, "A Framework for the Automatic Geometric Repair of CityGML Models," *Cartogr. from pole to pole*, pp. 187–202, 2014.
- [94] N. Alam, D. Wagner, M. Wewetzer, V. Coors, and M. Pries, "Towards automatic validation and healing of CityGML models for geometric and semantic consistency," in *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2013, vol. II, no. November, pp. 27–29.
- [95] H. Ledoux, M. Meijers, K. Arroyo Ogori, and J. Zhao, "Sessie 5: Validation and Repair of 2D & 3D Data," in *OSGeo. nl dag 2013, Delft, The Netherlands, November 13, 2013*, 2013.
- [96] V. Coors and M. Krämer, "Integrating quality management into a 3D geospatial server," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 2011, vol. 38, no. 4C21, pp. 7–12.
- [97] D. Wagner and H. Ledoux, "OGC CityGML Quality Interoperability Experiment," 2016.
- [98] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. 2010.
- [99] A. Gentle, "Continuous integration and delivery for documentation," 2015. [Online]. Available: <https://opensource.com/business/15/7/continuous-integration-and-continuous-delivery-documentation>.
- [100] H. Vakharia, "Git, Docker, and continuous integration for TeX documents," 2015. [Online]. Available: <https://opensource.com/business/15/12/git-docker-continuous-integration-tex-documents>.
- [101] I. Prieto, J. L. Izgara, A. Egusquiza, and R. Béjar, "Sustainable refurbishment in urban districts through a web-based tool based on 3D city model," *Sustain. Places*

- 2015, p. 31, 2015.
- [102] S. Chacon and B. Straub, *Pro Git. Section 5.1.*, 2nd ed. Apress. Berkely, CA, USA., 2014.
- [103] A. M. Berg, *Jenkins Continuous Integration Cookbook - Second Edition*, Second edi. Packt Publishing Ltd, 2015.
- [104] U. Isikdag and S. Zlatanova, “Interactive modelling of buildings in Google Earth : A 3D tool for Urban Planning,” *Dev. 3D Geo-Information Sci.*, pp. 52–70, 2010.